EXPLOTAR VULNERABILIDADES DE IOT CON GNURADIO Y SDR
IOT EXPLOITATION WITH GNURADIO AND SDR

ELIZABETH RIVERA ARELLANO

MÁSTER EN INTERNET DE LAS COSAS

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

TRABAJO FIN DE MÁSTER EN INTERNET DE LAS COSAS

CURSO 2019-2020
CONVOCATORIA: SEPTIEMBRE 2020
CALIFICACIÓN: 7,5 (NOTABLE)

DIRECTORES:

GUILLERMO BOTELLA JUAN
JOAQUIN RECAS PIORNO

Resumen

El auge de las comunicaciones inalámbricas sigue manteniéndose, llegando a formar parte de nuestra cotidianidad con el objetivo de hacerla más fácil, como por ejemplo mediante el uso de dispositivos wereable, la domótica, entornos industriales, entre otros que conforman el Internet de las Cosas.

Muchas veces, los fabricantes de esos dispositivos por mejorar su funcionalidad y hacerlos más competitivos en el mercado dejan a un lado el aspecto de la seguridad. Y nosotros como usuarios desconocemos lo viable que puede llegar a ser interceptar la comunicación de nuestros dispositivos IoT al usar un medio tan promiscuo como el aire para la comunicación.

El objetivo de este trabajo es mostrar la utilidad de herramientas como SDR y GNU Radio para estudiar vulnerabilidades en dispositivos IoT que emplean protocolos de Radio Frecuencia (RF), mostrando su frecuencia y modulación, decodificando la señal y reproduciendo paquetes de radio.

Palabras clave

IoT, SDR, GNU Radio, Replay, Ingeniería Inversa, RF, Modulación, HackRF One, RTL_SDR

Abstract

Wireless communications still keep growing, becoming part of our daily lives with the aim of making it easier. For example through the use of wireless devices, home automation, industrial environments, among others that make up the Internet of Things.

Sometimes, manufacturers of IoT devices focusing to improve their functionality and make them more competitive in the market and do not pay sufficient attention to the security issue. Normally, the common user is not aware of the vulnerabilities of communications that are propagated by the air as used by IoT devices.

This work pretends to show the useful of SDR and GNU Radio as tools to study and analyzing vulnerabilities over IoT comunnications through Radio Frequencies protocols (RF), getting interesting information such as operation frequency, modulation, applying reverse engineering and replay attack.

Keywords

IoT, SDR, GNU Radio, Replay, Reverse Engineering, RF, Modulation, HackRF One, RTL_SDR

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

En primer lugar quiero agradecer a Dios por haberme permitido llegar a donde me encuentro, y por contar con una familia y amigos extraordinarios que de una u otra forma me han hecho posible avanzar satisfactoriamente en este paso profesional. Asimismo, quiero agradecer a Guillermo Botella y Joaquín Recas por su acompañamiento durante la ejecución de este TFM.

Capítulo 1

Introducción

1.1. Motivación

El Internet de las Cosas (IoT) cada vez tiene mayor presencia tanto en el entorno industrial como en nuestra cotidianidad. Resulta muy común disponer en el hogar de dispositivos inteligentes que contribuyen a maximizar el grado de confort de las personas que allí habitan; así mismo, es común el uso de dispositivos wearables (equipos tecnológicos que se pueden llevar encima) para satisfacer ciertas necesidades.

En algunas ocasiones, por satisfacer la demanda del mercado de disponer gran variedad de alternativas de dispositivos IoT, se deja de lado aspectos como la seguridad. Considerando que gran parte de esos dispositivos utilizan la banda de frecuencia destinada para uso Industrial, Científico y Médico (ISM) pueden resultar vulnerables aquellas comunicaciones que utilizan radiofrecuencia para señales de alcance corto. Aunado a ello, las limitaciones computacionales de muchos de ellos dificultan que posean métodos que permitan robustecer el nivel de seguridad de estos.

Las comunicaciones inalámbricas facilitan muchas actividades en distintos entornos, pero también es una puerta a ataques maliciosos mediante el análisis del espectro de frecuencia. La intención de este trabajo es mostrar el uso de SDR y GNU Radio como herramientas potenciales para detectar vulnerabilidades en las comunicaciones de distintos dispositivos IoT.

1.2. Objetivos

El objetivo principal del proyecto es mostrar las potencialidades de SDR y GNU Radio para detectar vulnerabilidades en las comunicaciones entre dispositivos IoT mediante el análisis de las ondas electromagnéticas de distintos protocolos que utilizan el aire como medio de transmisión.

Esta prueba de concepto se llevará a cabo mediante los siguientes escenarios, que representan los objetivos específicos:

- Familiarización con las herramientas GNU Radio y SDR mediante la implementación de un receptor FM.
- Interceptar una señal de 433 MHz para efectuar Ingeniería Inversa y un ataque de Replay. Esta banda ISM es muy utilizada para comunicaciones a corta distancia y que requieran poco consumo de potencia como es el caso de un sistema domótico, en sistemas de telemetría para drones y otros vehículos de aeromodelismo, entre otros.

1.3. Estructura de la memoria

El documento está distribuido en seis capítulos diferenciados entre sí, conteniendo lo siguiente en los capítulos posteriores. El segundo capítulo contiene el estado del arte. El tercer capítulo condensa la información teórica relevante para la aplicación del entorno experimental. El cuarto capítulo describe el desarrollo del proyecto. En el quinto capítulo se presenta el análisis de los resultados obtenidos. Finalmente, en el sexto capítulo se mencionan las conclusiones y el trabajo futuro que se podría efectuar.

Atendiendo a la normativa, los capítulos 7 y 8 corresponden a la traducción al idioma inglés de los capítulos de introducción y conclusiones.

Capítulo 2

Estado del arte

El internet de las cosas (IoT) sigue presentando gran importancia en el ámbito tecnológico. Los analistas de 451 Research, empresa de investigación de la industria tecnológica, estiman que en 2020 hay 8.800 millones de dispositivos conectados en el mundo - sin contar consolas, ordenadores personales y televisores inteligentes-, cifra que podría alcanzar los 13.800 millones en 2024 . De acuerdo a la consultora IDC (International Data Corporation), el uso que hacen del IoT los españoles ha aumentado un 67 % [24].

Un ejemplo de la expansión de IoT se evidencia en la Tabla 2.1, que corresponde al reporte mostrado por IDC en Marzo del 2020 donde reseñan el comportamiento de la adquisición de dispositivos wearables y la proyección de la tasa de crecimiento anual compuesto (CAGR) dentro de 5 años [27].

Worldwide Wearables Shipments, Market Share, and Five-Year CAGR) 2020 and 2024 (shipments (ventas) in millions)					
Product	**2020 Shipments**	**2020 Share**	**2024 Shipments**	**2024 Share**	**2020—2024 CAGR**
Earwear/Hearables	203.8	55.4 %	301.5	57.2 %	10.3 %
Watches	95.0	25.8 %	149.5	28.4 %	11.4 %
Wristbands	65.1	17.7 %	69.8	13.3 %	1.8 %
Others	204.3	1.2 %	6.0	1.1 %	8.7 %
Total	368.2	100.0 %	526.8	100.0 %	9.4 %

Tabla 2.1: *Seguimiento trimestral mundial de dispositivos wearable, Marzo 2020 [27]*

Los dispositivos IoT por tener su variante inalámbrica utilizan el aire para propagar la información mediante ondas electromagnéticas (Fig 2.1). Gran parte de las tecnologías IoT operan en la banda ISM (Industrial, Médica, Científica). Esta banda es no licenciada y a pesar de tener amplio un abanico de frecuencias para transmitir, gran parte de los dispositivos se centran en frecuencias por debajo de 1 GHz:

- Banda ISM 433.92 MHz: Se extiende por Europa, África, Rusia y toda la geografía de la región 1 de la ITU (Unión Internacional de Telecomunicaciones). En esta banda interactúan la mayoría de dispositivos de bajo coste sin contar con seguridad en cuanto a la transmisión de datos [26].

- Banda ISM 868 MHz: Usada por las regiones que forman parte de la agrupación CEPT (Conferencia Europea de Administraciones de Correos y Telecomunicaciones) como Europa, Australia, Canadá, Estados Unidos, Perú, Nueva Zelanda, Antillas Neerlandesas. Israel y Sudáfrica. La distancia de cobertura es menor que la anterior pero cuenta con bajo nivel de perturbación en la transmisión [26].

- Banda ISM 915 MHz: Es el equivalente a la banda de 433.92 MHz para Estados Unidos y los países de las Américas pertenecientes a la región 2 de la ITU, además de Israel y Australia [26].

Adicionalmente, otras bandas muy usadas son las de 2.4 GHz y 5 GHz.

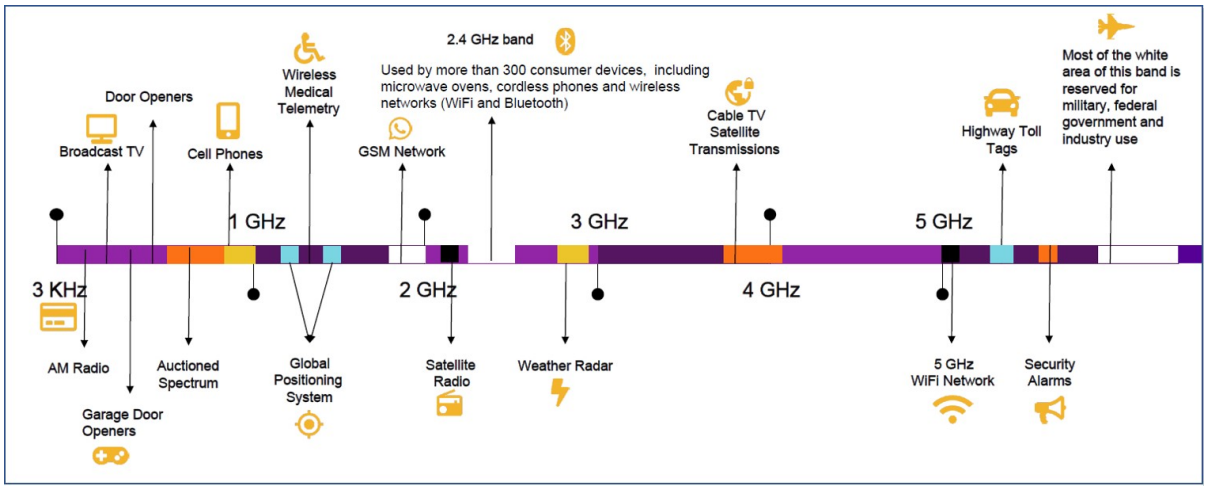


Figura 2.1: *Dentro del espectro de ondas de radio [4].*

Algunas de las tecnologías que transmiten en la banda ISM son:

- **Bluetooth**: La tecnología Bluetooth es un estándar de comunicaciones inalámbricas de corto alcance que opera en la banda no licenciada de 2.4 GHz y cuyo esquema de modulación es GFSK (Gaussian Frequency Shift Keying) [9].

- **WiFi**: Es una tecnología que emplea el estándar IEEE 802.11, opera en las bandas de 2.4 GHz, 5 GHz y 60 GHz con alcance moderado pero con un elevado consumo y ancho de banda [12]. Utiliza técnicas de modulación como BPSK, QPSK o QAM [13].

- **Zigbee**: Formado por protocolos basados en el estándar IEEE-802.15.4 que define el nivel físico y control de acceso para bajas tasas de transmisión de datos de redes inalámbricas de área personal (LR-WPAN). Forma parte de las bandas ISM y en Europa se utiliza la frecuencia alrededor de 868 MHz, en Estados Unidos la de 915 MHz, aunque la banda más popular es la de 2.4 GHz que no cuenta con limitación geográfica [13]. Utilizan las modulaciones BPSK ó O-QPSK, dependiendo de la capa física que se emplee [8].

- **LoRa**: Tecnología diseñada para la creación de redes de comunicación de bajo consumo o LPWAN (Low Power Wide Area Network). La distribución de frecuencias varía según

el país. En Europa se emplean las frecuencias comprendidas entre 868 MHz y 869 MHz, mientras que en Norte América van entre 902 MHz y 928 MHz. La modulación empleada es Chirp SS (Chirp Spread Spectrum) o frecuencia modulada pulsada de espectro ancho la cual es similar a FSK en cuanto a la variación de frecuencia [10].

El tema de seguridad siempre ha sido un punto neurálgico al momento de implantar soluciones IoT, sobre todo considerando la promiscuidad de un medio de propagación como lo es el aire.

Los ataques en redes IoT están incrementándose y el impacto de un ataque exitoso podría acarrear daños [25]. Dispositivos como SDR (Software Defined Radio) cada vez cobran más fuerza en el estudio de comunicaciones inalámbricas y, junto con GNU Radio como herramienta para el tratamiento de señales, han contribuido a la realización de distintos trabajos que demuestran la vulnerabilidad a nivel de la capa física de protocolos empleados en IoT.

En la GRCon17, conferencia de GNU Radio realizada anualmente, Matt Knight y Marc Newlin presentaron un ataque del tipo Jamming vulnerando el sistema de seguridad de una casa mediante la interferencia de la comunicación entre los sensores dispuestos en puertas y ventanas y el panel de control respectivo [3]. Para ello generaron una señal de ruido de 345 MHz a través de GNU Radio y colocando una jaula de Faraday para bloquear la señal entre los sensores y el panel de control.

En la DEF CON 23, una de las convenciones de hackers que existe, Samy Kamkar demostró que con un dispositivo electrónico de bajo coste se podía capturar señales del mando de un coche [2].

En el paper [21] se reseñan algunas pruebas de penetración en dispositivos Zwave escuchando los paquetes Zwave provenientes de un sensor magnético.

Capítulo 3

Marco Teórico

La tecnología de radio cada vez cobra mayor importancia en el IoT, por lo tanto no debe ignorarse los aspectos de seguridad que pueden acarrear desde la intervención de un canal inalámbrico hasta el control de la señal transmitida por el mismo. En este capítulo se tratan los conceptos relevantes en el ámbito de las pruebas de concepto realizadas en este trabajo, a saber:

- Tecnología SDR
- GNU Radio
- Bluetooth Low Energy
- Comunicaciones 433/315 MHz
- Módulo RF 433 MHz
- Placa Arduino
- Modulación
- Métodos de ataques inalámbricos

3.1. Tecnología SDR

3.1.1. Descripción General

La Radio definida por software (SDR) no es un concepto nuevo. Las primeras implementaciones data de los 90 en proyectos militares. El principio básico de un SDR es digitalizar la señal de radiofrecuencia (RF) lo más cerca posible de la antena, según las capacidades de los conversores analógicos-digitales. Esta señal RF digitalizada, es entregada a un ordenador u otro dispositivo de procesamiento para realizar el post-procesamiento de la señal, convirtiendo todos los elementos anteriormente analógicos en funciones ejecutadas (modulación, control automático de ganancia, reducción del ruido, entre otros) por un procesador digital [17].

La Fig 3.1 muestra un diagrama de flujo de procesamiento típico de SDR, donde resalta:

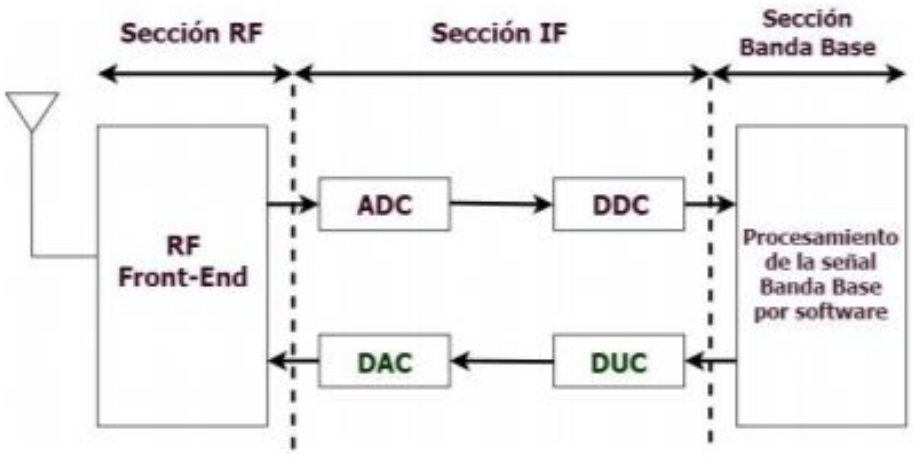


Figura 3.1: *Diagrama de flujo de un SDR [17].*

- Sección RF: Responsable de transmitir/recibir las señales de RF adecuándolas y convirtiéndolas a señales de frecuencia intermedia (IF), o bien, amplificarlas o modularlas. La amplificación generalmente se hace mediante amplificadores de bajo ruido (LNA) los cuales, en la etapa de recepción incrementan el nivel de amplitud de señal. A su vez, eliminan la mayor cantidad de ruido proveniente del canal. En la transmisión,

sirven para adecuar la potencia de la señal a un valor apto para su envío al canal.

- Sección IF: Sirve de puente entre el dominio analógico y digital. En ella se encuentran los módulos analógicos digital o digital analógicos (ADC/DAC) y los bloques de digital up/down converter (DDC/DUC). Aquí, en la parte de la recepción, la señal de IF pasa a Banda Base. El proceso inverso ocurre para el caso de la transmisión.

- Sección Banda Base: En el caso de transmisión se encarga de extraer la señal digitalizada en banda base para su posterior tratamiento a través de software. En el caso de recepción ocurre lo contrario.

3.1.2. Algunos usos de SDR

Desde la aparición de SDR, la industria de las comunicaciones inalámbricas ha avanzado a gran velocidad. Gracias a que SDR permite la creación y actualización de productos, la hace una tecnología con capacidad de customización. Algunas aplicaciones son: [28]:

- En el ámbito académico y de investigación permite el estudio de algoritmos de procesamiento de señales inalámbricas y nuevos protocolos de comunicación.

- Algunas start-up los emplean para el desarrollo de pruebas de concepto para los dispositivos.

- Como plataforma experimental con fines educativos.

- Es la herramienta preferida de los radioaficionados. Algunos aficionados a la radioastronomía construyen sus propios observatorios radioastronómicos con SDR.

- Fuera del ámbito educativo, los usuarios más comunes de los SDR son los hackers, las empresas de seguridad, las autoridades de monitoreo de radio y los laboratorios militares.

3.1.3. Hardware SDR

En este apartado se hará énfasis en los SDR utilizados para las pruebas de concepto efectuadas (véase 4).

- RTL-SDR: Los dongles RTL-SDR se diseñaron originalmente para DVB-T (Digital Video Boadcasting – Terrestrial) Recepción de HDTV (Televisión de alta definición), pero hoy en día son usados como SDR de propósito general por algunos hackers [1].

  Basado en el chip RTL2832U, es un receptor que oscila entre 24 MHz y 2.2 GHz. El ADC (Convertidor Analógico Digital) del RTL2832U ha adoptado un muestreo bidireccional I/Q de 8 bits, la frecuencia de muestreo más elevada para que funcione sin pérdida de datos es de 2.56 MS/s [28]. Es una excelente alternativa para aquellas personas que quieran iniciarse en la tecnología SDR. La imagen de un RTL-SDR se puede apreciar en la Fig 3.2.

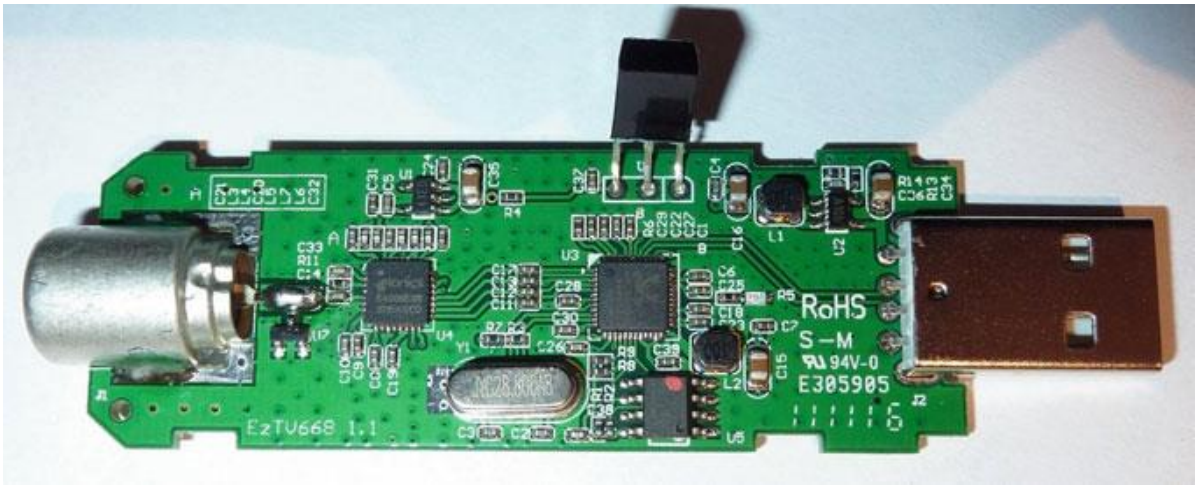


Figura 3.2: *RTL-SDR [28].*

- HackRF One: Es un SDR capaz de transmitir o recibir señales de radio desde 1 MHz hasta 6 GHz y puede alcanzar un sampling rate hasta 20 MS/s. Es un SDR de código abierto que puede ser usado como transmisor o receptor pero no al mismo tiempo ya

que trabaja en modalidad half duplex [28]. En la Fig 3.3 se muestra la imagen de este tipo de dispositivo.

Figura 3.3: *HAckRF One [17].*

Existen otras alternativas de SDR en el mercado. En la Tabla 3.1 se muestra un resumen de las características principales de los mismos.

	HackRF One	Ettus B200	Ettus B210	BladeRF x40	RTL-SDR	LimeSDR
Frequency Range	1MHz-6GHz	70MHz-6GHz	70MHz-6GHz	300MHz-3.8GHz	22MHz-2.2GHz	100KHz-3.8GHz
RF BW	20MHz	61.44MHz	61.44MHz	40MHz	3.2MHz	61.44MHz
Sample Depth	8 bits	12 bits	12 bits	12 bits	8 bits	12 bits
Sample Rate	20MSPS	61.44MSPS	61.44MSPS	40MSPS	3.2MSPS	61.44MSPS (Limited by USB 3.0 data rate)
Transmitter Channels	1	1	2	1	0	2
Receivers	1	1	2	1	1	2
Duplex	Half	Full	Full	Full	N/A	Full
Interface	USB 2.0	USB 3.0	USB 3.0	USB 3.0	USB 2.0	USB 3.0
Programmable Logic Gates	60 macro-cell CPLD	75k	100k	40k(115k avail)	N/A	40k
Chipset	MAX5864, MAX2837, RFFC5072	AD9364	AD9361	LM56002M	RTL2832U	LMS7002M
Open Source	Full	Schematic, Firmware	Schematic, Firmware	Schematic, Firmware	No	Full
Oscilator Precision	+/-20ppm	+/-2ppm	+/-2ppm	+/-1ppm	?	+/-1ppm initial, +/-4ppm stable
Transmit Power	-10dBm+, (15dBm @ 2.4GHz	10dBm+	10dBm+	6dBm	N/A	0 to 10dBm (depending on frequency)

Tabla 3.1: *Características de algunos SDR [28].*

3.2. GNU Radio

Es un framework de software libre y código abierto que permite el procesamiento de señales mediante el uso de bloques. Las aplicaciones de GNU Radio están escritas en Pyt-

hon pero las funciones implementadas a bajo nivel utilizan C++. La Fig 3.4 muestra la arquitectura software de GNU Radio.

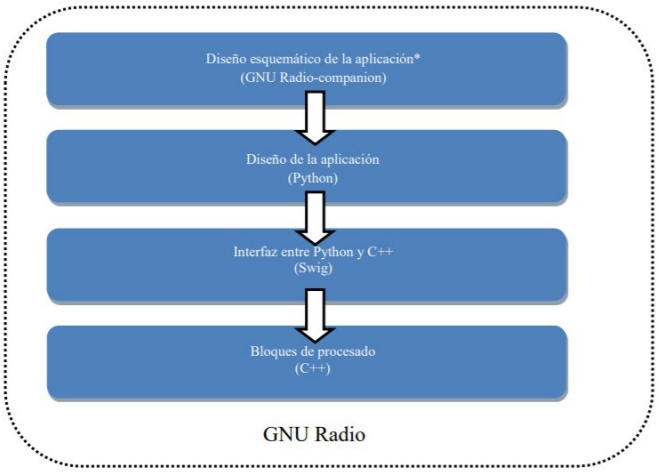


Figura 3.4: *Arquitectura GNU Radio [15].*

Dispone de una interfaz gráfica llamada GNU Radio Companion (GRC) que permite la creación de flujogramas para el análisis de señales a partir de bloques incluidos en librerias o customizados por el usuario.

Los bloques utilizados para la construcción de módulos a través de GRC se pueden agrupar de la siguiente manera [14]:

- Sources (Fuentes): Especifican cualquier tipo de fuente tales como ficheros, un micrófono, hardware radio.
- Bloques de procesado de señal: Acá se consideran filtros, moduladores, demoduladores, remuestreadores, amplificadores, etc.
- Sinks (Sumideros): En el sink la señal apuntará a un fichero, la tarjeta de sonido, hardware radio, visualizadores gráficos como FFT, etc.

GNU Radio Companion (GRC) está conformado por (Fig 3.5):

- Workspace: Es la zona de diseño donde se construye el flujograma. Por defecto contiene el bloques Options que permite el ajuste de algunos parámetros generales del diagrama de flujo y el bloque Variable que permite el ajuste de la frecuencia de muestreo.
- Library: Se tiene un listado de los bloques instalados y disponibles en GRC ordenados por categorías.
- Terminal: Se registran los mensajes relacionados a el estado de ejecución de un flujograma.
- Variables: Muestra información de las variables contenidas en el flujograma.

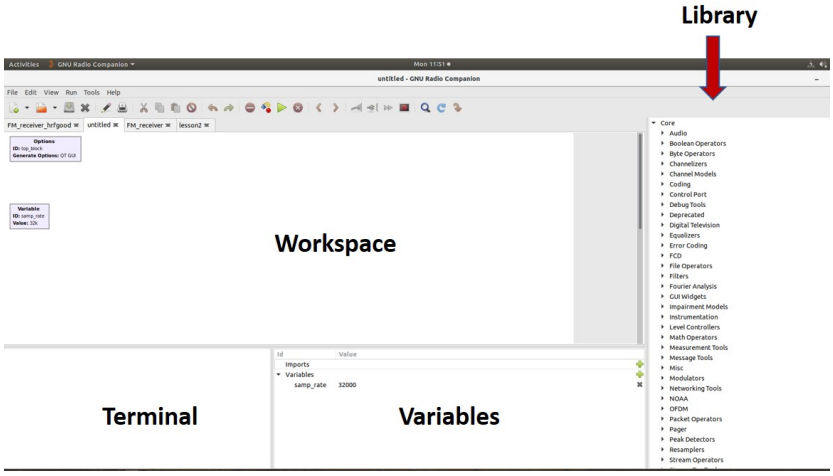


Figura 3.5: *Estructura GRC*

Soporta un amplio rango de hardware SDR y es extensible ya que permite crear un bloques en particular dependiendo del requerimiento que se tenga. Es muy útil para propósitos de investigación a nivel de seguridad, aplicaciones de negocio y para las personas aficionadas a las señales de radio.

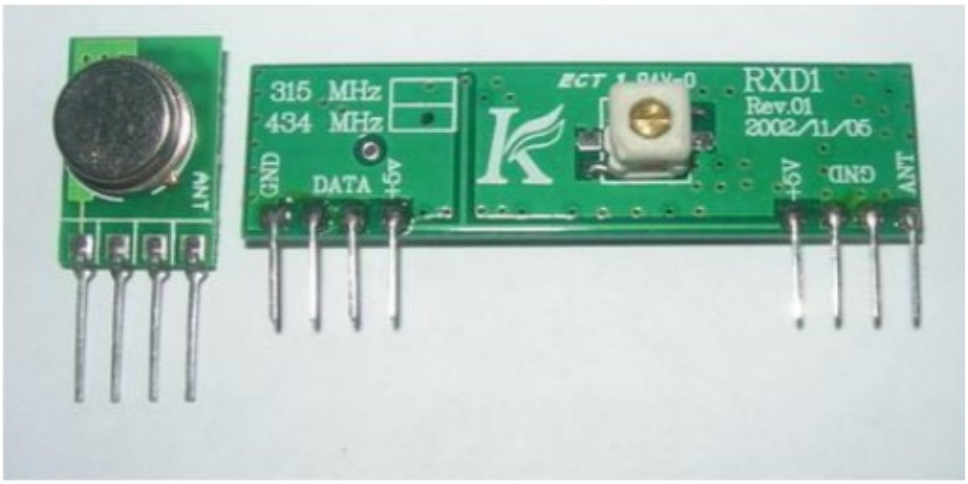
3.3. Comunicaciones 433/315 MHz

La banda ISM de 433/315 MHz es muy utilizada por dispositivos de baja potencia para comunicaciones a corta distancia donde los protocolos de comunicaciones por lo general son

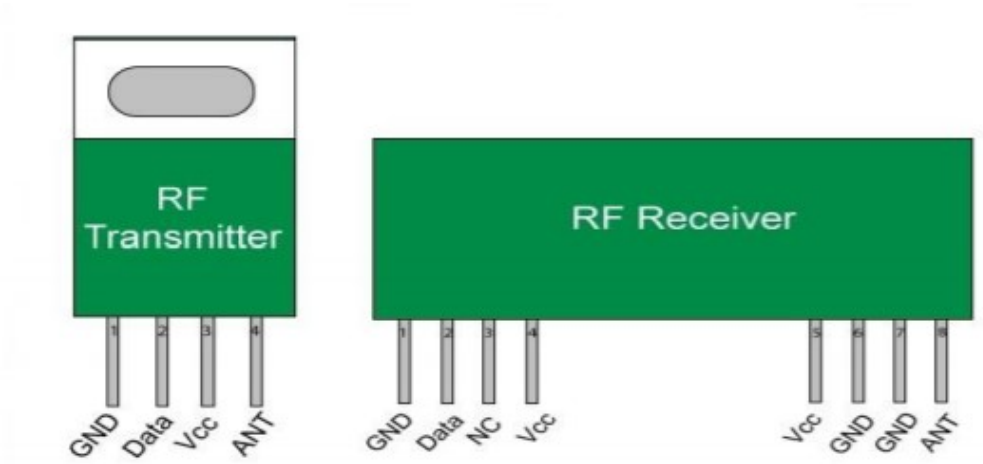
simples y por ende tienden a ser vulnerables. Esta banda es muy utilizada por dispositivos empleados en la cotidianidad como control remoto de luces, de puertas de garaje, de juguetes, entre otros. Adicionalmente, algunos sensores empleados en IoT utilizan chips RF que trabajan en esa misma gama de frecuencias [28].

3.4. Módulo RF 433 MHz modelo FS1000A

El módulo de RF (Fig 3.6) consta de un transmisor y un receptor RF que operan a 434 MHz. Un transmisor RF recibe la data en serie y la transmite de forma inalámbrica mediante su propia antena a una velocidad de transmisión que oscila entre 1Kbps y 10 Kbps. El receptor RF opera en la misma frecuencia [7].



(a) Módulo RF



(b) Pinout módulo RF

Figura 3.6: *Módulo RF 433 MHz [7].*

3.5. Placa Arduino

Arduino es una de las placas de desarrollo de código abierto que existe. Una de sus versiones, arduino nano, está basada en el microcontrolador ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x), con 14 pines GPIO digitales y 8 pines analógicos y puede ser colocada en una Protoboard [16]. Es muy utilizado en diversos proyectos tanto en el mundo IoT como en la electrónica, la Fig 3.7 corresponde al Arduino nano empleado en

este trabajo.

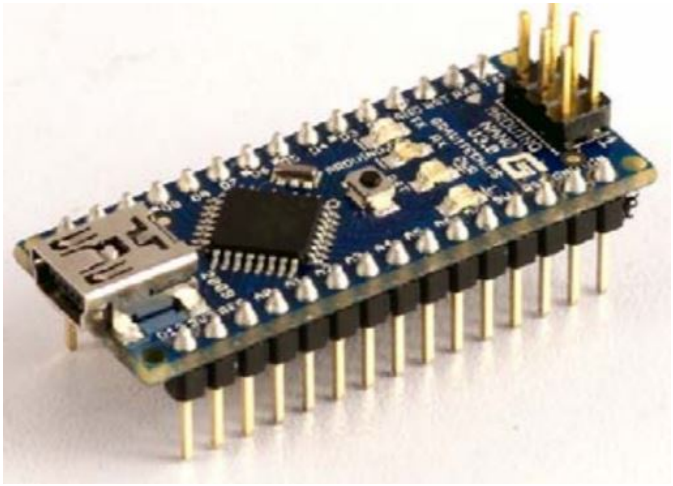


Figura 3.7: *Arduino Nano [16].*

3.6. Modulación

Para que un flujo de datos se transmita como ondas de radio debe ser modulado o convertido del dominio digital al dominio de radiofrecuencias. En este punto se hablará de las modulaciones que son consideradas básicas y de las cuales surgen diferentes variaciones empleadas por tecnologías IoT, pero manteniendo su principio.

3.6.1. Modulación ASK

El transmisor de 433 MHz emplea la modulación por desplazamiento de amplitud (ASK - Amplitude Shift Keying) O también conocida como OOK (On-Off-Keying) que consiste en variar la amplitud de la señal portadora entre los valores 1 y 0 según la señal moduladora. Es una variación de la modulación AM [18].

La función de la señal modulada ASK viene dada por [23]:

$$f_{ASK}(t) = f(t) * cosW_c t \tag{3.1}$$

Donde f(t) es una señal binaria con la información a transmitir y la señal portadora está definida por cosWct. La Fig 3.8 muestra el resultado de la modulación ASK.

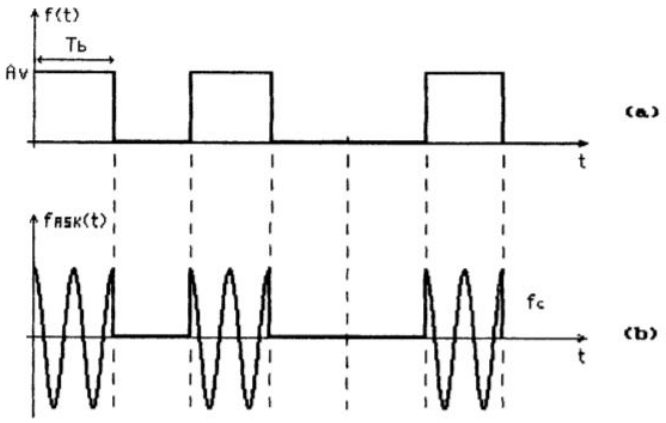


Figura 3.8: *Señal modulada en ASK: (a) Señal binaria de información; (b) Señal modulada ASK [23].*

3.6.2. Modulación FSK

La modulación por desplazamiento de frecuencia (FSK) se emplea en transmisiones digitales donde la señal moduladora varía entre dos valores de tensión discretos que forman un tren de pulsos representados por un 1 y un 0 [19].

Los datos de entrada binaria modifican a la frecuencia de centro de portadora, lo que origina que cuando el dato de entrada sea 1 la señal obtenida tenga una frecuencia mayor y si es 0 su frecuencia será menor tal como se evidencia en la Fig 3.9.

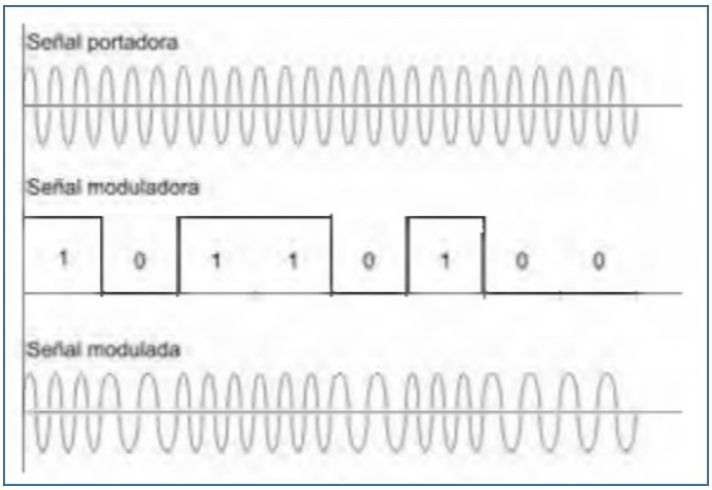


Figura 3.9: *Modulación FSK [19]*

.

Las siguientes funciones algebraicas permiten obtener los valores de las frecuencias alta y mínima, y la frecuencia en cada momento temporal [6]:

$$F0(t) = Ac * cos(2\pi(FC - \Delta f)t) \tag{3.2}$$

$$F1(t) = Ac * cos(2\pi(FC - \Delta f)t) \tag{3.3}$$

Donde FC hace referencia a la frecuencia portadora, Ac a la amplitud de la portadora y Δf al desplazamiento máximo de la frecuencia.

En la Fig 3.10 se muestra el espectro de una señal modulada en FSK.

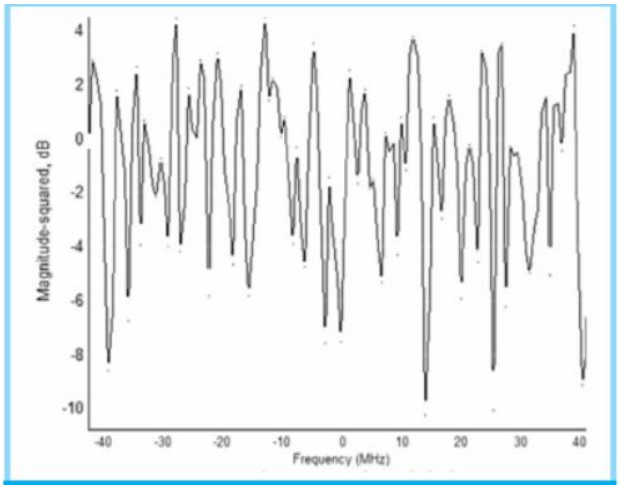


Figura 3.10: *Espectro de una señal modulada en FSK [11].*

3.6.3. Modulación BPSK

La modulación por desplazamiento de fase (Phase Shift Keying - PSK) es un esquema de modulación digital donde se envía mensajes al detectar cambio de fase de la señal portadora, posee un bajo índice de error [20].

Una de sus variantes es la modulación por desplazamiento de fase binaria (Binary Phase Shift Keying - BPSK), en donde los valores de la señal de entrada digital (1 y 0) son convertidos a una señal NRZ (Not Return to Zero) dipolar (1 y -1) y la fase de la portadora es asignada de 0 a π [6].

La fase de la portadora se modifica de manera proporcional a la señal de entrada, expresándose como sigue [6]:

$$C(t) = Ac * cos(2\pi * FC * t + \phi_c) \tag{3.4}$$

Siendo ϕ_c la fase de la portadora.

Luego ocurre un cambio de fase de la onda portadora entre 0 y 180 grados, dando como reultado:

$$S_{psk} = Ac * cos(2\pi * FC * t + \phi_c + \pi * m(t)) \tag{3.5}$$

Al ser la fase inicial de la portadora 0, $\phi = 0$:

$$S_{psk} = Ac * cos(\pi * m(t)) * cos(2\pi * FC * t) \tag{3.6}$$

P(t) que es la señal de entrada puede tomar los valores 1 y -1. Es multiplicada por la señal portadora:

$$P(t) = cos(\pi * m(t)) \tag{3.7}$$

La señal modulada en su forma algebraica quedaría como sigue:

$$S_{psk}(t) = P(t) * C(t) = P(t) * Ac * cos(2\pi * FC * t) \tag{3.8}$$

En la siguiente Fig 3.11 se muestra un ejemplos de la modulación BPSK

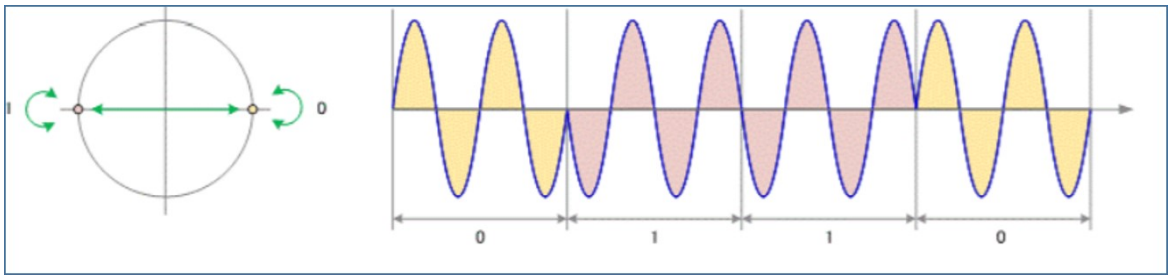


Figura 3.11: *Ejemplo BPSK [6].*

3.7. Algunos métodos de ataque inalámbrico

3.7.1. Sniffing

Se puede considerar como una observación pasiva del tráfico que está siendo transmitido, donde se logra recolectar información relevante del mismo y que puede servir para complementar otros ataques ofensivos.

3.7.2. Ingeniería Inversa

Este tipo de método permite al atacante visualizar la señal y procesarla. La visualización de la señal se logra grabando la misma y para ello es requerido que el atacante disponga de cierta información del objetivo transmisor y de un hardware SDR compatible que permita interceptar la señal. El procesamiento de la señal implica determinar aspectos como el esquema de modulación empleado y decodificar o determinar la información contenida en la señal transmitida [22].

3.7.3. Replay

Los ataques de Replay son muy comunes en las comunicaciones RF, y consiste en retransmitir la señal capturada a un receptor. Para llevar a cabo este tipo de ataque, no se requiere disponer de mucho conocimiento del dispositivo objetivo o de la señal que se está propagando, lo único que se debe conocer es la frecuencia en la que opera el dispositivo destino [22].

3.7.4. Jamming

Similar a un ataque de denegación de servicio (DoS), este tipo de ataque es muy popular y su objetivo es bloquear la señal mediante la transmisión de una señal de interferencia para que el receptor no reciba la señal esperada. Es usualmente usado junto con ataques del tipo 3.7.3. Suelen usarse para bloquear la comunicación crítica entre sistemas que dependen de la misma para su correcto funcionamiento como por ejemplo el control del tráfico aéreo y las aeronaves [22].

3.7.5. GPS Spoofing

El objetivo de este tipo de ataques son las señales de GPS no cifradas y no autenticadas enviadas por satélites a cualquier dispositivo que tenga GPS habilitado. Funciona comparando la señal GPS que recibe un dispositivo y luego aumenta lentamente la amplitud de la señal

GPS falsa hasta que el dispositivo comienza a responder a la nueva transmisión GPS como si ésta fuera legítima, de éste modo el atacante puede redirigir los vehículos a su ubicación deseada. Este tipo de ataques está dirigido principalmente a barcos y aeronaves [22].

3.7.6. Man-in-the-middle

Este tipo de ataque corrompe la integridad y confidencialidad de las sesiones. El atacante utiliza información de la red para hacerse pasar por algún recurso de ésta [5].

Capítulo 4

Desarrollo del proyecto

Los dispositivos IoT no están exentos de ser vulnerables, En este capítulo se describen los componentes o herramientas a nivel de hardware y software que se emplearon para la elaboración de las pruebas de concepto sobre dos escenarios de comunicaciones inalámbricas, así como detalles de las mismas.

4.1. Herramientas de Hardware y Software

4.1.1. Herramientas Hardware

- Receptor RTL-SDR: Se utiliza el dongle RTL2832U que funciona como receptor para señales de frecuencia 24 MHz a 2.2 GHz (véase 3.1.3).
- HackRF One: Utilizado para recibir y transmitir señales en el rango de frecuencia de 1 MHz a 6 GHz (véase 3.1.3).
- Arduino Nano: Utilizado como microcontrolador.
- Módulo transmisor RF 433 MHz: Usado para transmitir/recibir la señal de 433 MHz con el microcontrolador arduino.

4.1.2. Herramientas Software

- GNU Radio: Framework que permite el procesamiento de señales de radio.

- GQRX: Es una herramienta que permite analizar visualmente el espectro de frecuencia de una señal.
- Audacity: Programa gratuito y de código abierto para analizar señales de audio.
- Arduino IDE: Entorno de desarrollo para arduino.

4.2. Entorno de pruebas

El propósito de las pruebas planteadas es generar un soporte que avale lo versátil que puede llegar a ser SDR y GNU Radio para interceptar señales electromagnéticas que utilizan muchos de los dispositivos que forman parte de nuestra cotidianidad.

Aprovechando el kit de aprendizaje de explotación de IoT de Attify adquirido por la Universidad, se utilizó el mismo como punto de referencia y de partida para la interacción con SDR y GNU Radio.

El entorno de pruebas se centra en los siguientes escenarios:

- Escenario 1: Implementa un receptor FM.
- Escenario 2: Intercepta una señal RF de 433 MHz.

4.2.1. Receptor FM

Una de las utilidades más comunes de un SDR es la implementación de un receptor de frecuencia modulada (FM), lo cual es una excelente alternatuva para familiarizarse en el entorno SDR - GNU Radio. En este apartado se implementa un receptor FM con GNU Radio para visualizar de forma gráfica el espectro FM entre un rango de frecuencias especificado y enviado posteriormente a una tarjeta de audio. La Fig 4.1 esquematiza este escenario.

El primero paso es recibir la señal de radio FM y para ello se emplea el bloque Osmocom Source que contiene los drivers de diversos SDR y que está incluido en todas las distribuciones de GNU Radio. Éste tendrá como entrada el dongle RTL_SDR con una frecuencia

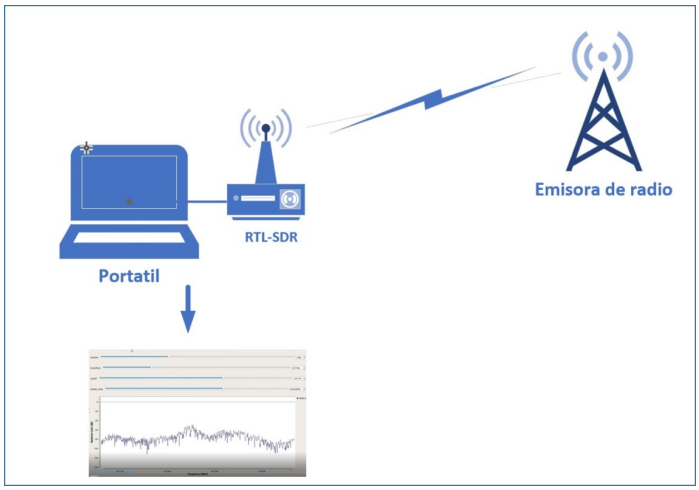
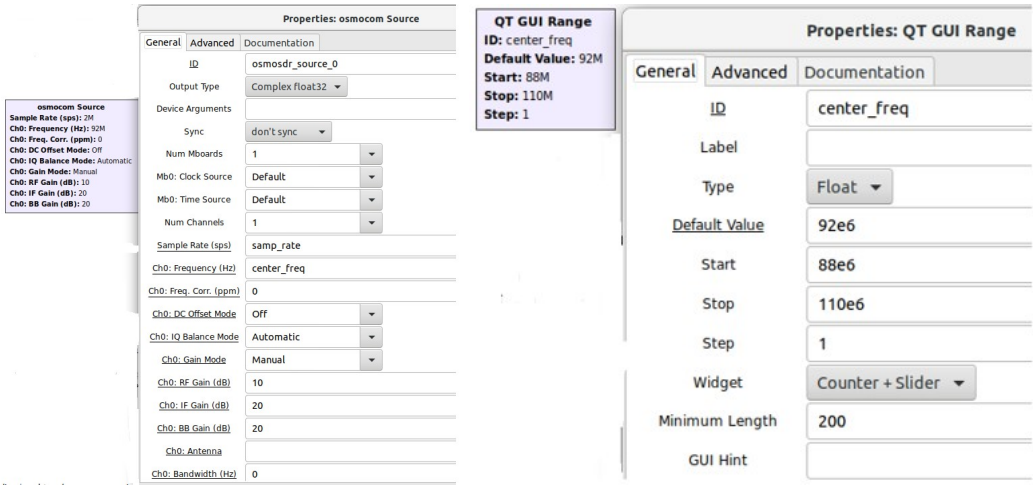


Figura 4.1: *Diagrama Receptor FM*

de muestreo de 2 millones de muestras por segundo, ver Fig 4.2(a). La frecuencia central es considerada una variable para hacerla ajustable, de modo que su configuración se realiza mediante un bloque QT GUI Range tal como se evidencia en la Fig 4.2(b).



(a) Bloque Osmocom Source

(b) Bloque QT GUI Range para el rango de la frecuencia central

Figura 4.2: *Configuración del Osmocom Source y la frecuencia central*

Lo siguiente es efectuar el procesamiento correspondiente para que llegue a la tarjeta de audio de nuestro sistema cuya frecuencia de operación es de 48 KHz, por lo tanto hay que tener en cuenta que se parte con una frecuencia de muestreo de 2 MHz y se desea llegar al

destino con una frecuencia de 48 KHz. Para continuar con el procesamiento de la señal, ésta se limita de manera que sólo mantenga las frecuencias que sean de nuestro interés con ayuda de un filtro pasa bajo. Este filtro además de fijar la frecuencia central, decima la salida (en este caso toma 1 muestra de cada 4 muestras) para establecer la frecuencia intermedia (IF) en el receptor FM que para efectos de prueba se fijó en 4 permitiendo así disponer de 500 Kilomuestras por segundo (sample_rate/decimation = 2M/4). En la Fig 4.3 se muestran los parámetros a considerar en dicho bloque.

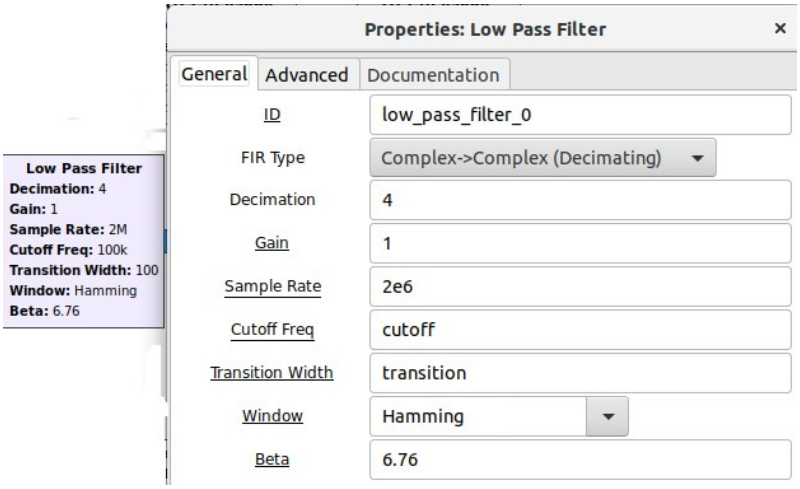
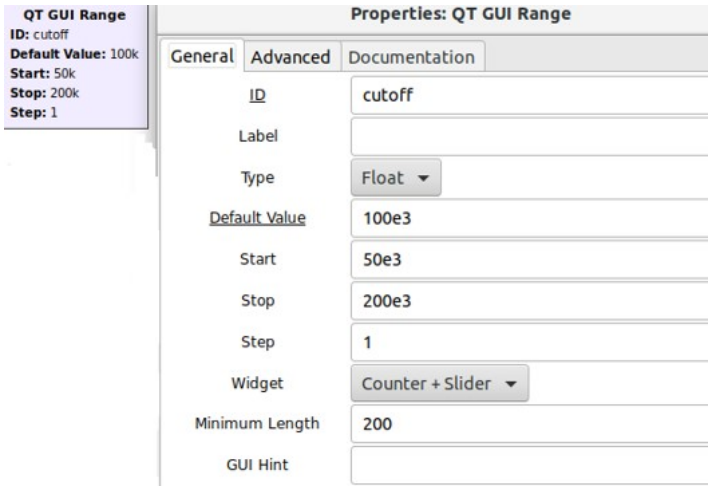
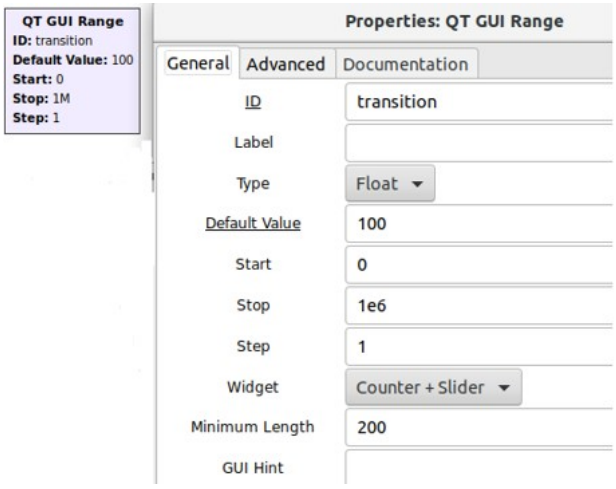


Figura 4.3: *Bloque Filtro pasa Bajo*

Para lograr que la frecuencia de corte (cutoff) y la transición (transition) en el filtro pasa bajo sea ajustable, estos se configuran como variables a través de bloques QT GUI Range para modificar los valores mediante un selector tal como se puede observar en la Fig 4.4.



(a) Bloque Multiply Const



(b) Bloque QT GUI Range para el rango de volumen

Figura 4.4: *Configuración del ajuste de la frecuencia de corte y la transición*

Esa señal es demodulada a través de un receptor FM ancho (WBFM o Wide Band FM Receive), ver Fig 4.5 al que por un lado ingresa información IQ y tiene como salida una señal del tipo float que corresponde al audio demodulado. Se usa como parámetro de cuadratura la tasa de muestreo que tiene como salida el bloque anterior, es decir 500K, y el parámetro de decimado se mantiene en factor 1. La frecuencia de muestreo a la salida de este bloque es de 500 KHz (Quadrature_rate / Audio Decimation).

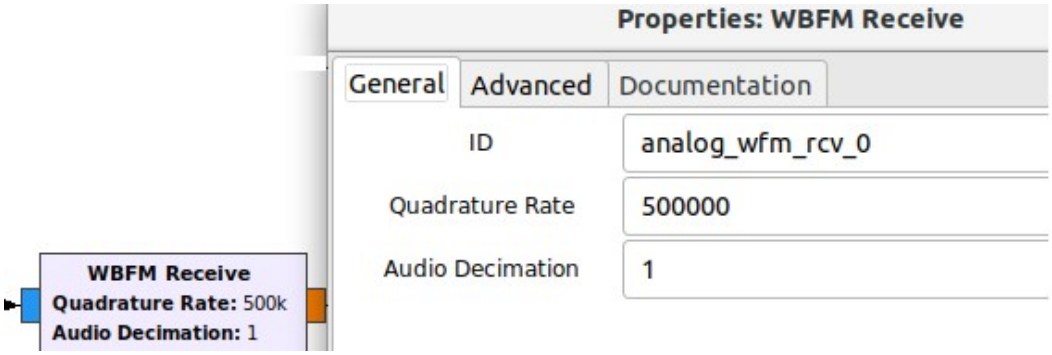


Figura 4.5: *Bloque Receptor FM*

En este punto la frecuencia de muestreo se mantiene en 500 KHz, a través del bloque Rational Resampler se reajusta el muestreo para así adecuarlo a lo requerido por la tarjeta de audio. Con los parámetros de interpolación y decimación se logra tal fin, y la interpretación sería algo como ingresa lo especificado en la decimación (500 KHz) y sale lo especificado en interpolación (48 KHz).La Fig 4.6 muestra su configuración.

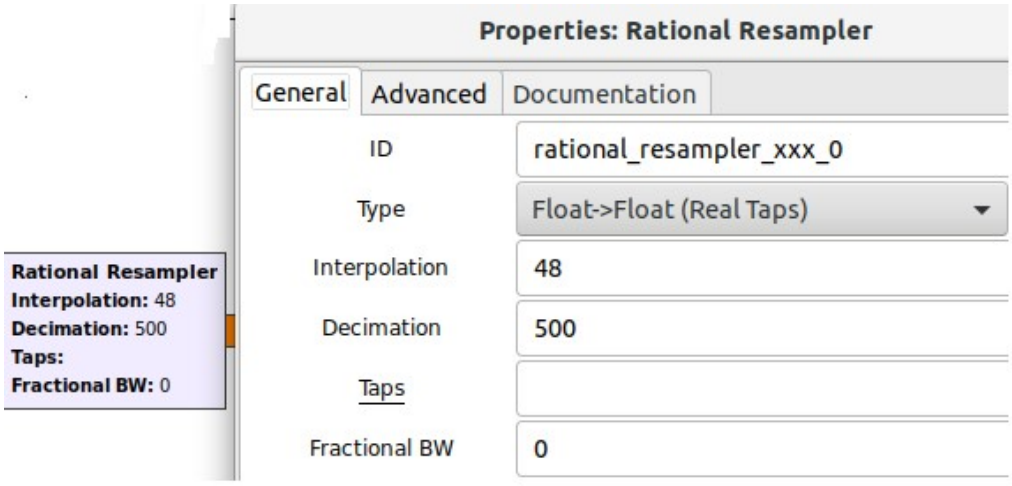
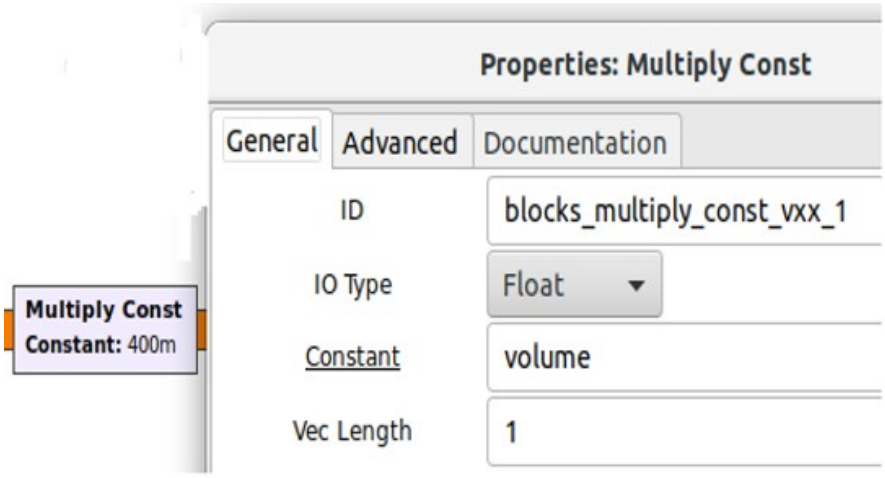
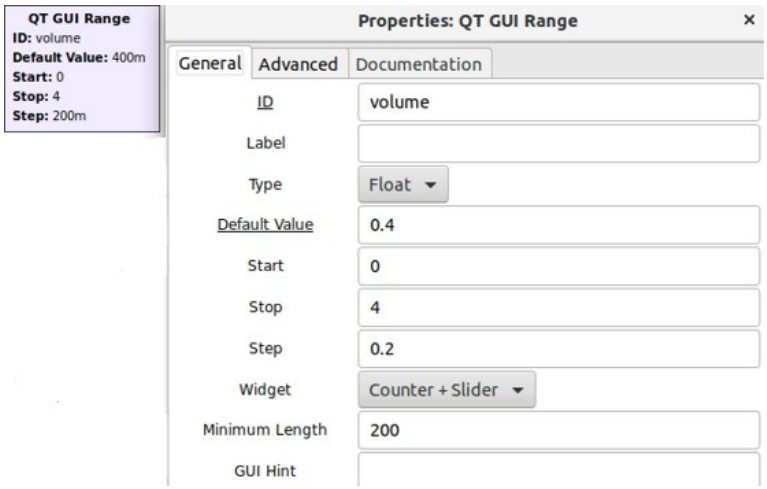


Figura 4.6: *Bloque Rational Resampler*

Para permitir tener control sobre el volumen del audio demodulado se emplea el bloque Multiply Const cuyo parámetro Constant se le asigna como volumen al que se le asocia un bloque QT GUI Range (elemento gráfico que permite definir rangos) con ese mismo nombre (volumen), ver Fig 4.7.



(a) Bloque Multiply Const



(b) Bloque QT GUI Range para el rango de volumen

Figura 4.7: *Configuración del control de volumen*

Finalmente, la señal procesada es enviada a la tarjeta de audio del computador mediante el bloque Audio Sink, Fig 4.8.

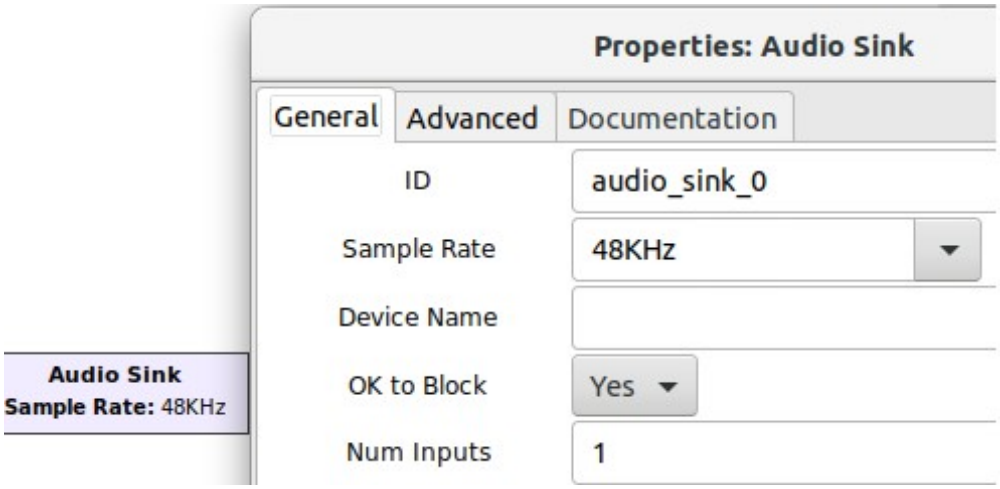


Figura 4.8: *Bloque Audio Sink*

El flujograma completo se muestra en la Fig. 4.9 y en la gráfica de la Fig 4.10 se visualiza el espectro de la frecuencia de la banda 102.3 MHz (una emisora de radio FM captada) con la información que está transmitiendo y que puede ser percibida por los altavoces de la computadora.

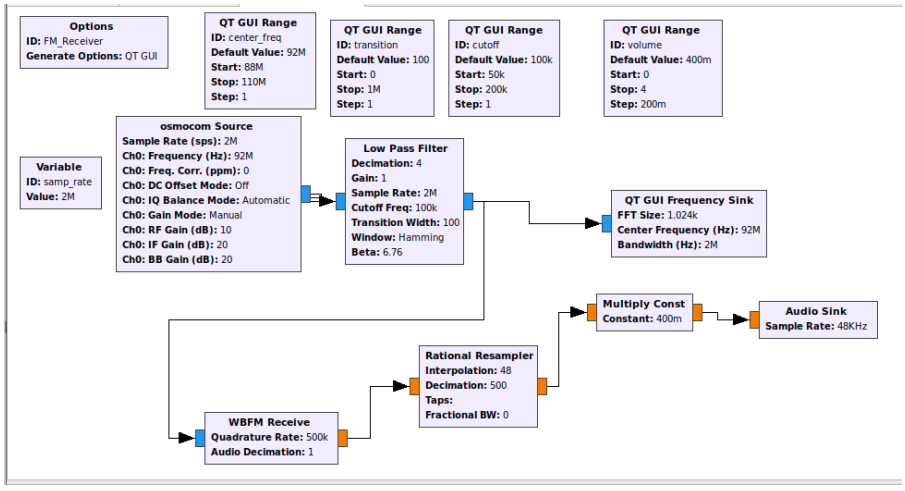


Figura 4.9: *Flujograma Receptor FM*

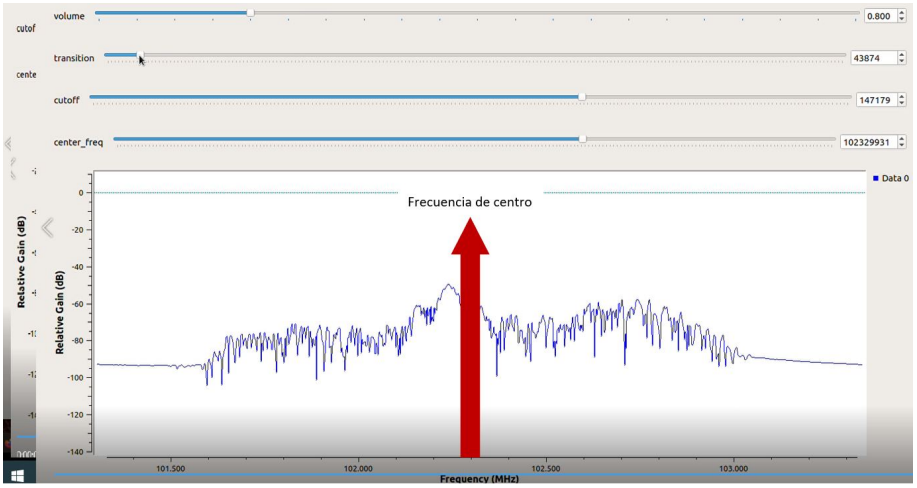


Figura 4.10: *Señal captada de la banda 102.3 MHz FM*

4.2.2. Análisis de vulnerabilidad en RF

El segundo escenario se subdivide en dos partes:

- Parte 1: Consta de dos microcontroladores (arduino nano y arduino uno) funcionando como cliente y servidor respectivamente. El cliente transmite una orden al servidor para encender y apagar un led mediante un dispositivo transmisor de 433 MHz. Aprovechando las bondades del RTL_SDR se intercepta la data, se almacena en un fichero .wav para posteriormente ser analizada y decodificada.La Fig 4.11 esquematiza lo mencionado anteriormente.

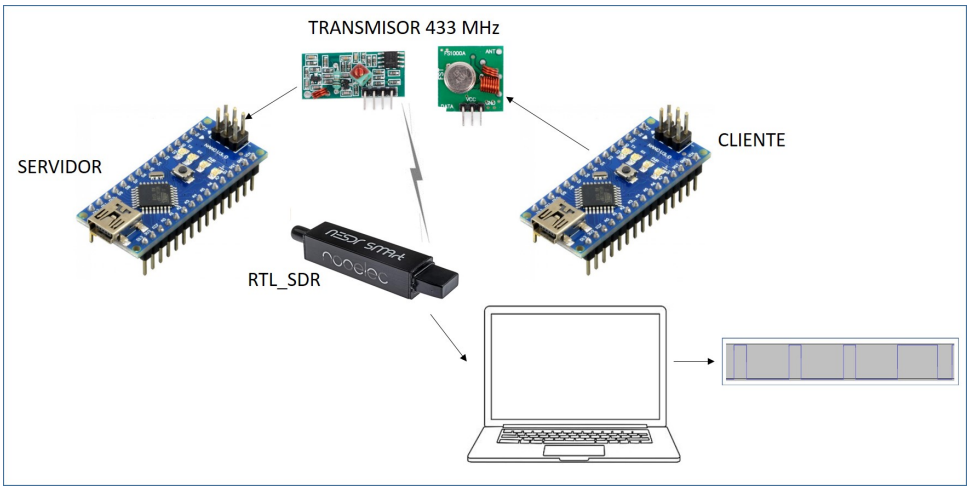


Figura 4.11: *Esquema Parte 1*

En primer lugar, se agrega a los arduino un código para transmitir y otro para recibir haciendo uso del IDE y de la libreria RC-Switch. El circuito utilizado para simular el escenario se muestra en la Fig 4.12.

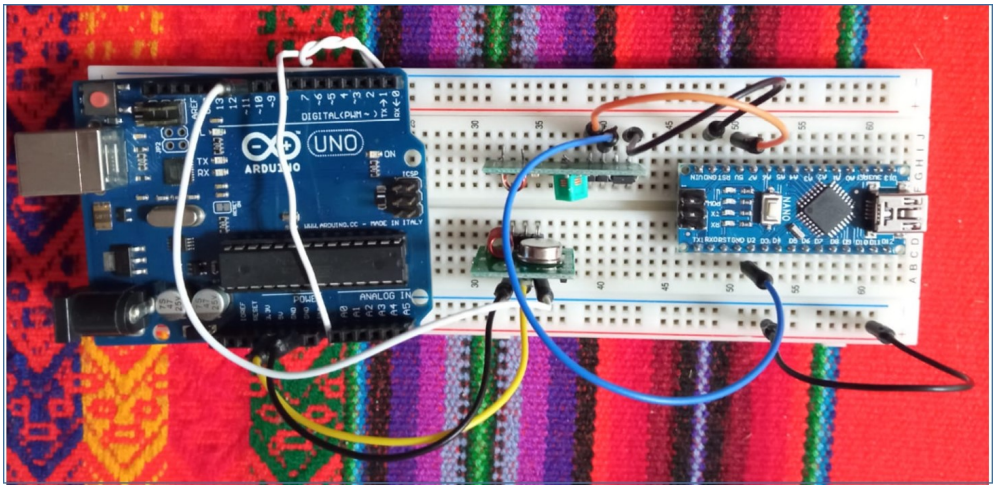


Figura 4.12: *Circuito escenario 1*

La frecuencia empleada por el transmisor es un dato ya conocido (433 MHz) y la modulación utilizada es del tipo ASK tal como se menciona en 3.4. En este punto se hará un paréntesis para comentar que en caso que no se disponga de dicha información, a través de GQRX se puede conocer la frecuencia. Para ello se observa el espectro

de frecuencia alrededor o cercano a 433 MHz (considerando que es una de las más usadas para IoT) identificando así la señal con más intensidad al ocurrir la transmisión (Fig 4.13).

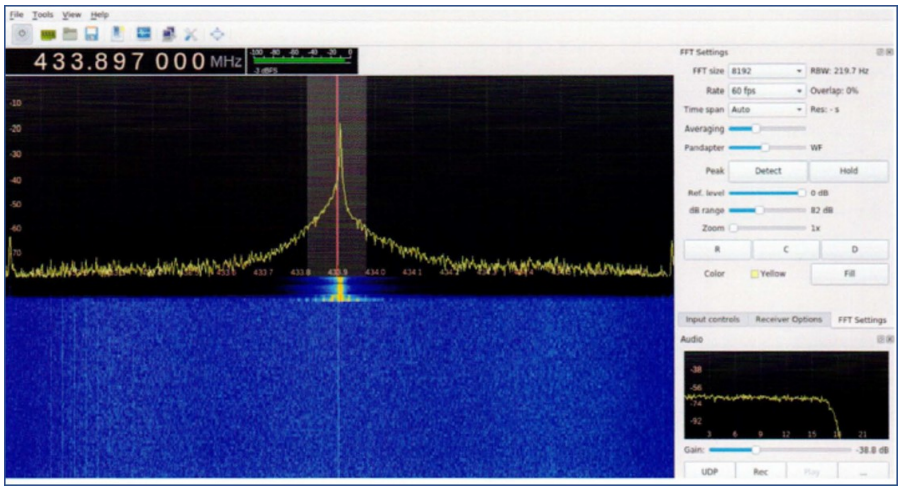


Figura 4.13: *Señal capturada mediante GQRX*

Otra opción viable para conocer parámetros como frecuencia, modulación y otra información de interés para el análisis de la comunicación es a través del FCC ID (Federal Communications Commission IDentifier) que es un identificador ubicado en muchas piezas de hardware. La Comisión Federal de Comunicaciones (FCC) se encarga de certificar los productos que se pueden vender en su territorio y para comprobar el cumplimiento de las normativas de dicha comisión se dispone del FCC ID. Mediante su página web `https://apps.fcc.gov/oetcf/eas/reports/GenericSearch.cfm` o a través del siguiente enlace `https://fccid.io/` se puede obtener documentación con información como la frecuencia y el tipo de modulación empleada.

Basado en lo antes expuesto se construye el flujograma correspondiente en GNU Radio. Para la recepción de la señal se utiliza el RTL_SDR mediante el bloque RTL_SDR Source diseñado justamente para solo SDR de ese tipo, al que se le fija como frecuencia 433 MHz.

El siguiente paso es demodular la señal AM recibida reduciendo el ruido y para ello

se hace uso del bloque llamado Complex to Mag. La salida del mismo es del tipo float lo que quiere decir que se ha eliminado la componente imaginaria de la señal lo que obliga a que los datos de los bloques subsiguientes sean del mismo tipo.

Para mejorar la amplitud de la señal que contiene los datos se utiliza el bloque Multiply Const y a través del bloque Threshold se convierte la señal a una del tipo rectangular. Este último bloque se usa como decisor, de acuerdo al umbral especificado se le asigna a la señal un 1 ó un 0.

Finalmente se obtiene la señal en formato .wav mediante el bloque Wav File Sink, que direcciona la señal recibida a la ubicación que se le asigne dentro del computador. Adicionalmente se puede monitorear en tiempo real con ayuda del bloque QT GUI Sink el cual permite visualizar el comportamiento de la señal en función de la frecuencia, del tiempo, su espectograma y constelación. El flujograma queda tal como se muestra en la Fig 4.14.

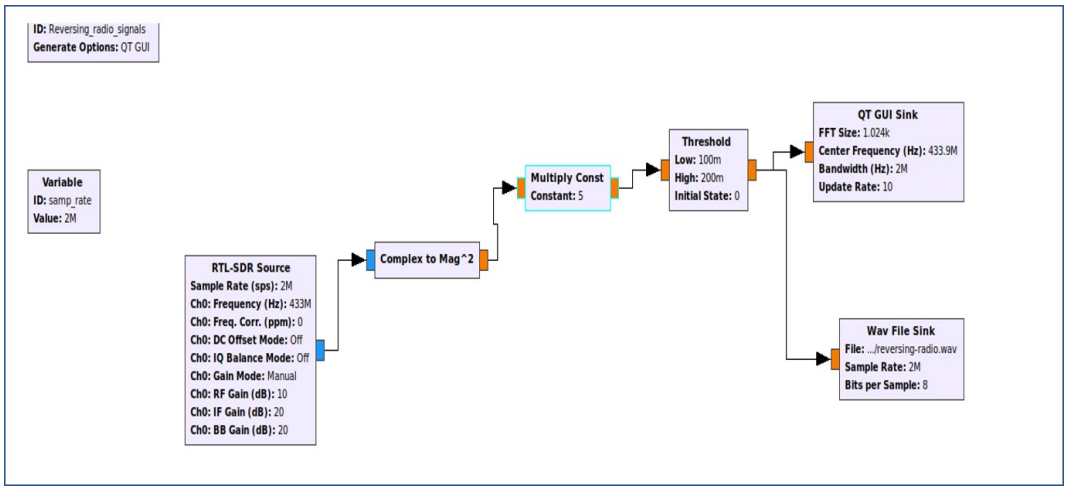
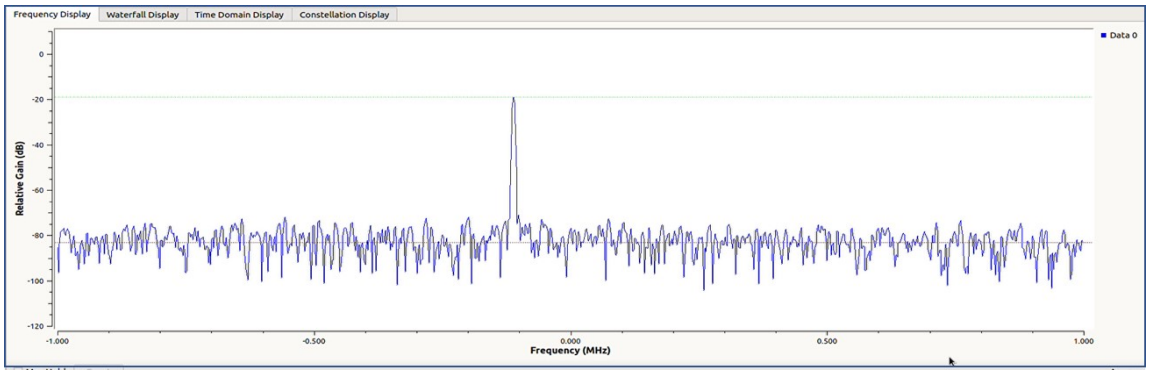


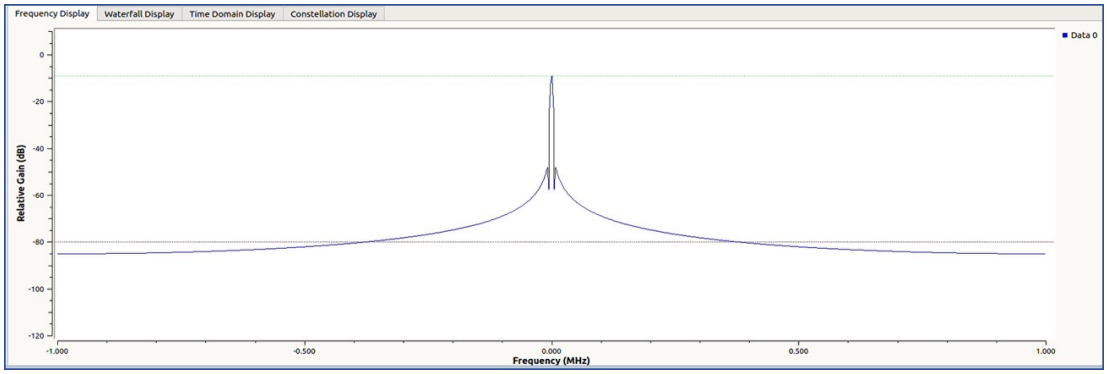
Figura 4.14: *Flujograma para capturar una señal de 433 MHz*

En las gráficas de las Fig 4.15 se evidencia el comportamiento de la señal captada antes y después de ser procesada en función de la frecuencia. La señal recibida por el SDR no sólo contiene la información de interés sino que está acompañada por mucho

ruido, propio de una banda no licenciada, en tanto que luego de ser procesada dicho ruido desaparece.



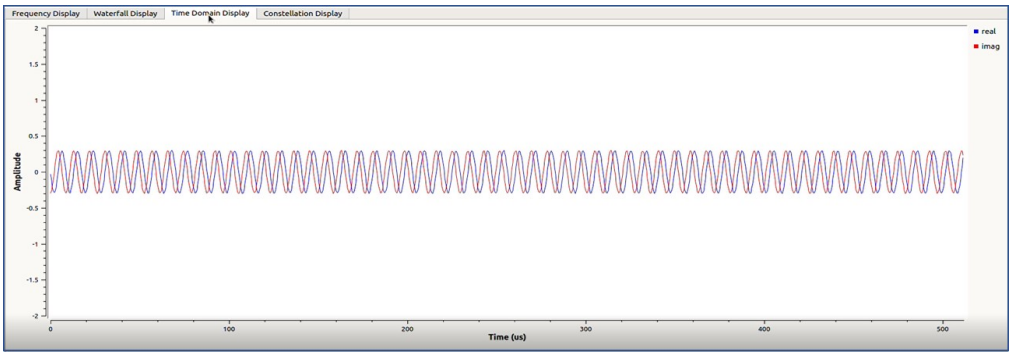
(a) Antes de ser procesada



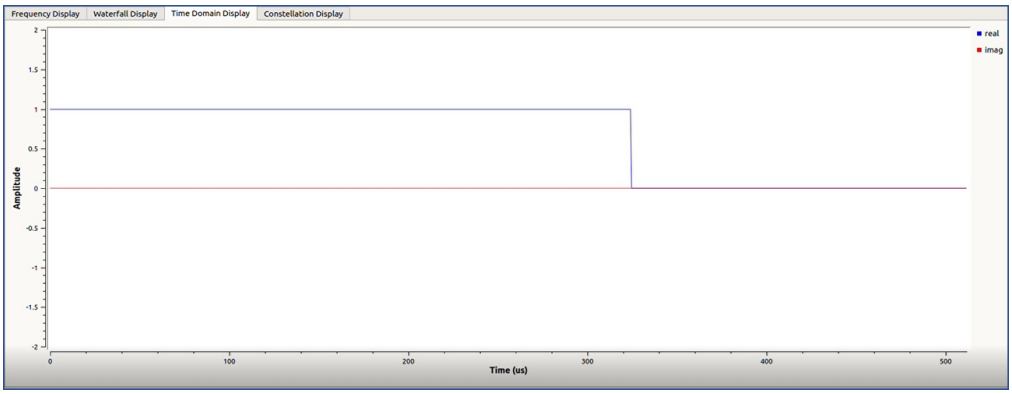
(b) Después de ser procesada

Figura 4.15: *Comportamiento de la señal en el dominio de frecuencia*

De manera análoga, la señal recibida presenta sus dos componentes real e imaginaria (Fig 4.16(a)) y luego de ser procesada es llevada a forma rectangular conservando solo su componente real (Fig 4.16(b)).



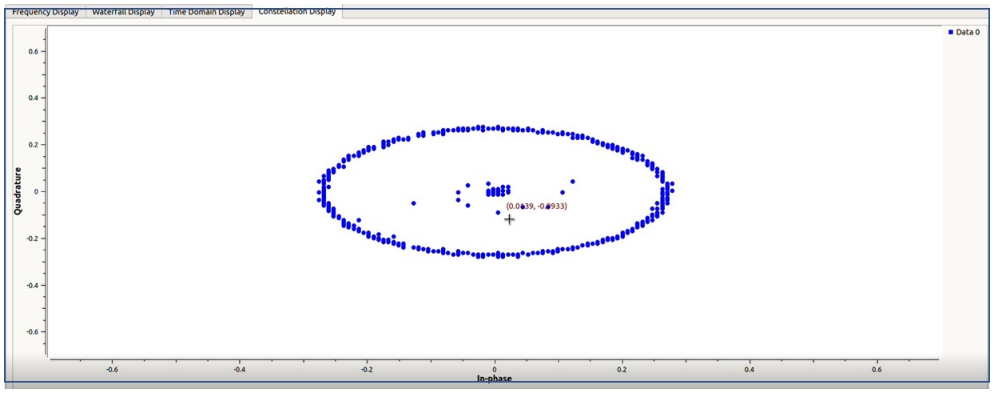
(a) Antes de ser procesada



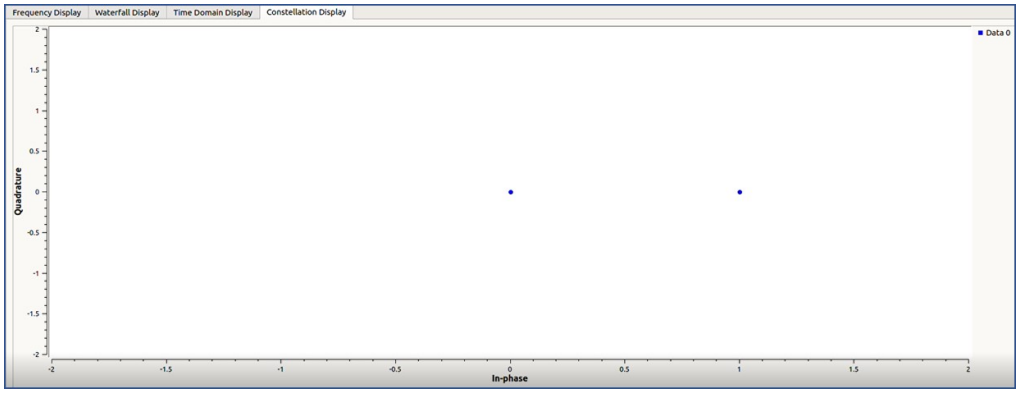
(b) Después de ser procesada

Figura 4.16: *Comportamiento de la señal en el dominio de tiempo*

En cuanto al diagrama de constelación, hay que tener en cuenta que tiene sentido en modulaciones donde la forma de onda asociada a todos los símbolos es una señal sinusoidal de la misma frecuencia pero con fase y amplitud variable de acuerdo a cada símbolo. La Fig 4.17(b) muestra la constelación que corresponde a una modulación ASK, dos puntos que se corresponde con cada uno de los símbolos de la modulación; el primero se sitúa en el origen y el segundo en el eje X con una amplitud determinada.



(a) Antes de ser procesada



(b) Después de ser procesada

Figura 4.17: *Constelación de la señal*

En este punto ya se cuenta con la señal grabada en formato wav y a través de la herramienta Audacity se puede analizar su contenido. La Fig 4.18 muestra la señal grabada visualizada en Audacity donde se evidencia que el tren de datos se envía cada dos segundos coincidiendo con lo programado en los arduino.

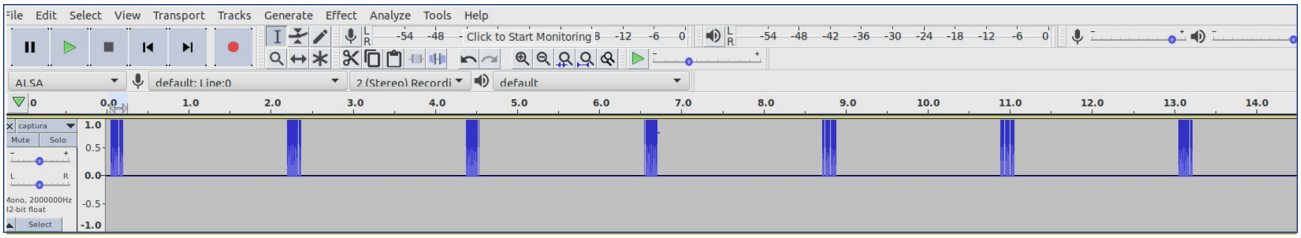
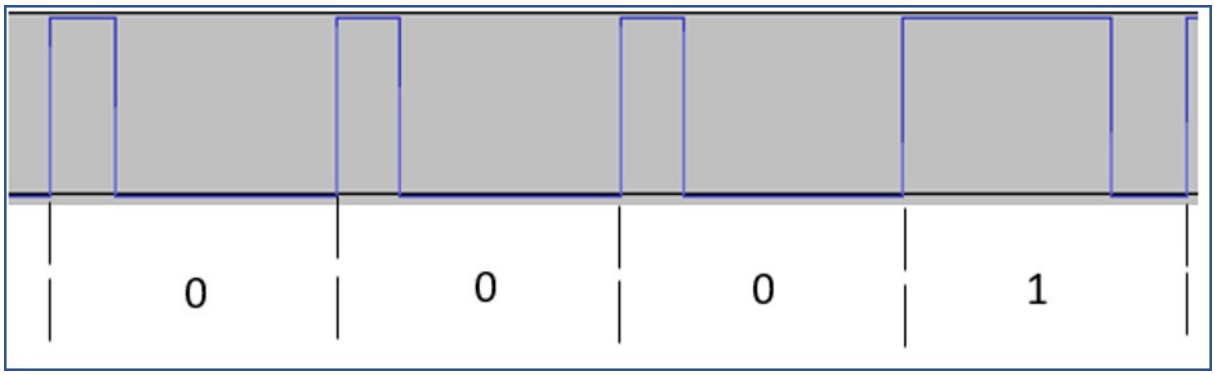
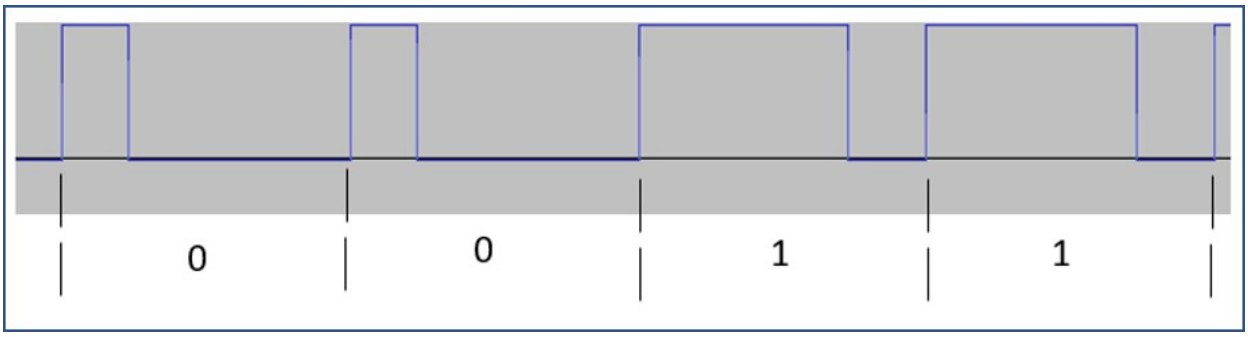


Figura 4.18: *Señal visualizada en Audacity*

Al hacer zoom sobre una de las repeticiones de cada orden se puede observar que su comportamiento corresponde a una modulación en amplitud, específicamente del tipo ASK ya que se registran pulsos de distintas duraciones. Para decodificar la señal se toma como referencia el valor que tenga a la mitad de un período, si está en alto se interpreta como un 1 y si está en bajo como un 0. En la Fig 4.19 se aprecia mejor lo comentado, que corresponde a las órdenes configuradas para apagar y encender un led.



(a) Señal OFF decodificada



(b) Señal ON decodificada

Figura 4.19: *Demodulación de la señal grabada*

- Parte 2: Haciendo uso del HackRF, se transmite la señal grabada de modo que el dispositivo receptor ejecute las órdenes contenidas en el tren de datos capturados (encender y apagar un led). La Fig 4.20 esquematiza lo mencionado.

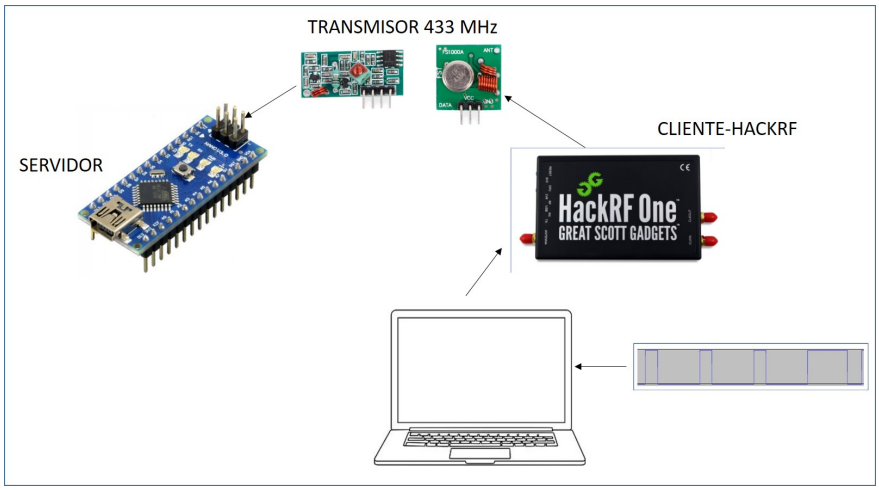


Figura 4.20: *Esquema Parte 2*

Para llevar a cabo este paso, a través de GRC se generan dos flujogramas, el primero de ellos para capturar la señal en formato IQ y el segundo para transmitir esa señal con el HackRF One.

El flujo empleado para la captura de la señal es similar al usado en la parte 1, con la diferencia que la información se almacena a través del bloque File Sink que permite llevar a un fichero las muestras I y Q capturadas por el SDR, dicho fluograma es mostrado en la siguiente Figura 4.21

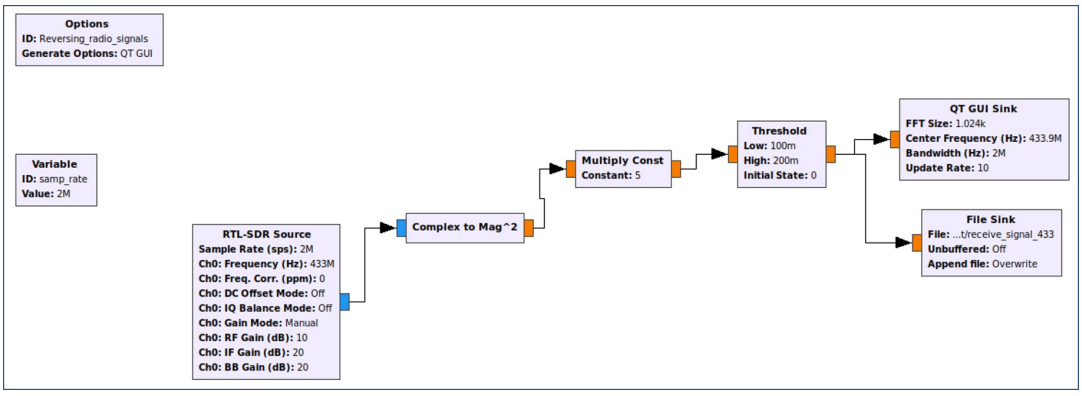


Figura 4.21: *Flujograma para grabar la señal de 433 MHz en formato IQ*

La transmisión de la información obtenida se realiza a través del HackRF One que

tiene la funcionalidad de transmitir además de recibir. El flujograma generado para ello toma la señal grabada en el fichero IQ a través del bloque File Source y lo entrega a un bloque Osmocom Sink que se encarga de enviar la información al periférico SDR. El flujo generado para tal fin es el que se visualiza en la Fig 4.22.

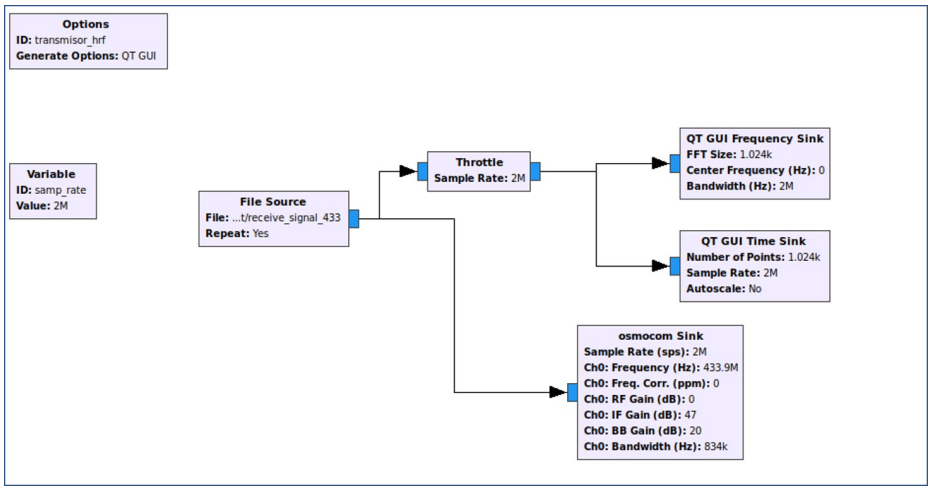


Figura 4.22: *Flujograma para transmitir la señal de 433 MHz grabada*

Se comprueba su funcionamiento eliminando del circuito indicado en la Fig 4.12 el microcontrolador arduino que funciona como cliente y al ejecutar el programa en GNU Radio el HackRF lo reemplaza transmitiendo el tren de datos que enciende y apaga un led en la placa que funciona como servidor.

Capítulo 5

Análisis de Resultados

En este capítulo se expondrán los resultados de los escenarios contemplados en el desarrollo del proyecto.

5.1. Receptor FM

Con la implementación de un receptor FM, más allá de haber sido el primer contacto con las herramientas objetos de estudio en este trabajo, se ha demostrado la flexibilidad de las mismas al permitir visualizar el espectro de frecuencia sin hardware adicional. Además, de facilitar programar distintas aplicaciones mediante una interfaz gráfica.

Aunque parezca trivial, es importante recordar que todo empieza en una fuente y termina en una salida que puede ser gráfica, audio, etc. Siendo la entrada una señal de tipo complejo y la señal de audio de tipo float con una tasa de muestreo menor que la empleada por el SDR, fué requerido aplicar el concepto de decimación para lograr tal reducción y así permitir que la tarjeta de audio procesara la señal recibida.

5.2. Análisis de vulnerabilidad en RF

Aunque en el contenido de éste apartado en el desarrollo del proyecto no se mencionó específicamente, muchos de los pasos efectuados corresponden a algunos de los ataques en redes inalámbricas que existe. Durante esa prueba se englobaron tres tipos de ataques a

saber:

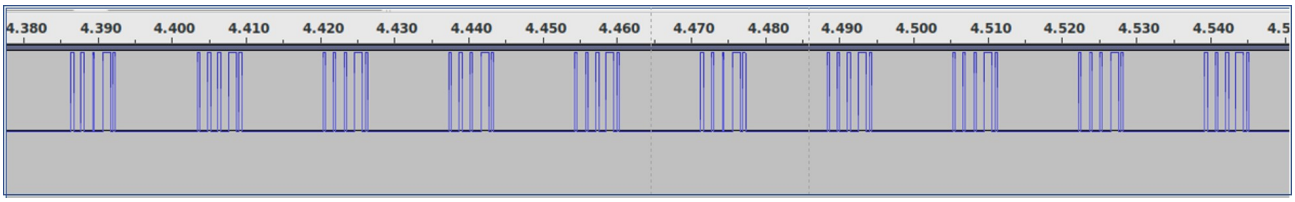
- **Sniffing:** Conocer información acerca de la frecuencia de operación de los dispositivos involucrados en la transmisión de datos, y llegar a entender el tipo de modulación implícita en la misma abarcaría lo concerniente al Sniffing. Esta información sirvió de base a los siguientes tipos de ataques que se mencionan a continuación.

- **Ingeniería Inversa:** Se cubrieron los dos pasos mencionados en 3.7.2, captura y procesamiento de la señal. Para capturar la data se recolectó cierta información del dispositivo transmisor (frecuencia de operación) y una vez capturada se identificó el esquema de modulación de la señal y el patrón de transferencia que posee como parte del procesamiento.

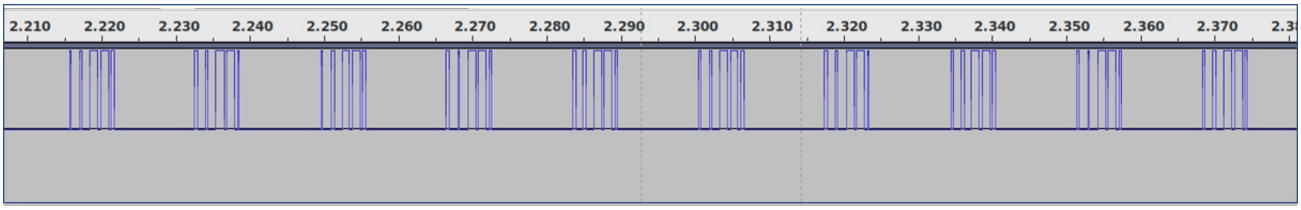
  El esquema de modulación ASK es el que manejan la mayoría de los dispositivos que transmiten en 433 MHz, que por imitar un patrón binario crean un comportamiento visible para determinados períodos de tiempo.

  Mediante mecanismos de decodificación, el atacante puede hacerse de la información transmitida que coincide perfectamente con lo realizado al obtener la información referida a la orden de encendido y apagado.Validando así que se puede decodificar ciertas señales capturadas desde el aire.

  El análisis realizado para decodificar la señal permitió validar que se transmitió múltiples repeticiones con la misma información. En la imagen de la Fig 5.1 se evidencian diez repeticiones de la señal de encendido (`0011`) y diez repeticiones de la señal de apagado (`0001`).



(a) Señal OFF



(b) Señal ON

Figura 5.1: *Información de la señal grabada*

El número de repeticiones fué programado en el microcontrolador arduino mediante el código para transmitir de la librería RC_Switch, específicamente con la siguiente sentencia `mySwitch.enableTransmit(10)`, donde el valor 10 corresponde al número de veces que se transmite un tren de datos.

En dispositivos comerciales como telemandos se puede presentar este comportamiento, lo cual lo hace un protocolo redundante ya que se requiere enviar varias veces la misma información para garantizar que la orden llegue al receptor. Este aumento en las repeticiones de envío origina disminución en la velocidad de transmisión, además de ocasionar interferencias al ocupar por más tiempo el espectro RF, que de por sí ya es un medio promiscuo. Esto también afecta el tiempo de vida útil de la batería del dispositivo.

- **Replay:** En el momento en el que el SDR HackRF transmite la señal previamente grabada al dispositivo receptor suplantando la identidad del dispositivo emisor original, se está llevando a cabo un ataque del tipo Replay. En este caso se trata de encender y apagar un led en un microcontrolador, pero esto se puede extrapolar a señales de mandos de garaje, interruptores inalámbricos ó tomar el control de un dispositivo a

control remoto.

El modelo de ataque de Replay mostrado en esta memoria funciona perfectamente para dispositivos que manejen código fijo; sin embargo, existen otros que emplean códigos variables que requieren más trabajo y un mayor análisis al momento de implementar este tipo de ataque.

Se puede resumir que para llevar a cabo un ataque de RF se debe pasar por las siguientes fases:

- Recopilación de información
- Identificar la frecuencia
- Conocer la modulación
- Transmitir la información

Además, es muy importante la frecuencia de muestreo. Cada flujograma generado en GNU Radio crea un archivo en python, por lo que se pueden llegar a construir sistemas interesantes teniendo nociones básicas de programación. En tanto que si se tiene un nivel mayor de experiencia a nivel de programación, se pueden elaborar bloques customizados para un fin determinado y de esa forma generar sistemas más potentes e incluso automatizar procesos.

Los códigos para cada uno de los escenarios se detallan en el apéndice A.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

Luego de la recopilación de información durante el desarrollo de este trabajo se puede concluir que aunque el concepto de Radio Definido por Software no es nuevo, es un hardware que puede adaptarse y funcionar con diferentes formatos de transmisión. Aunado a ello, el protagonismo que ha ganado IoT en la cotidianidad y en distintos sectores ha hecho de esta herramienta junto con GNU Radio una excelente alternativa para el análisis de distintos protocolos de comunicación como RF, Bluetooth, WiFi entre otros.

Así como se incrementa la cantidad de opciones IoT disponibles, también aumenta las posibles brechas a nivel de seguridad de los mismos. Siendo más sensibles los dispositivos IoT que operan en las bandas ISM para frecuencias sub-gigahercio al lograrse no sólo interceptar la transmisión sino comprometiendo la integridad de la comunicación al suplantar la identidad de un dispositivo.

La realización de las pruebas reseñadas en esta memoria fue un reto ya que el resultado esperado no se obtuvo en el primer intento sino que fue requerido efectuar múltiples repeticiones. Pero sin duda el hecho de que SDR y GNU Radio sean herramientas de código abierto y cuenten con una excelente comunidad que continuamente está actualizándose fue de gran ayuda.

Si bien es cierto que la dupla SDR – GNU Radio puede emplearse para vulnerar co-

nexiones RF, también puede brindar la opción de efectuar prototipos o hacking ético que permita el desarrollo de mecanismos de protección para robustecer la seguridad que sirvan de mecanismos de defensas ante los diferentes ataques planteados.

6.2. Trabajo Futuro

El trabajo realizado puede escalar a un entorno real por el simple hecho de tener este tipo de comunicaciones al alcance. Como trabajo futuro se propone tener en cuenta las siguientes consideraciones:

- El método de decodificación empleado en la prueba de concepto realizada es efectivo pero no muy eficiente. De hecho si la transmisión contiene un número de paquetes considerable resultaría algo complicado descifrar la señal, por tal razón sería interesante automatizar este proceso para que el mismo programa identifique el nivel correspondiente a un 1 y 0 lógico y así lograr una decodificación más expedita.

- Emplear las diversas contribuciones efectuadas sobre la vulnerabilidad de diferentes tecnologías IoT para desarrollar nuevos métodos de protección ante nuevas amenazas que puedan emerger.

Capítulo 7

Introduction

7.1. Motivation

Along time Internet of Things (IoT) has increased its presence both industrial environment and our daily lives. In most homes there are smart devices that maximize the confort of people that live there; likewise, the use of wearable devices has increased in order to satisfy certains needs.

Sometimes, in order to respond the market demand offering a wide of IoT devices, aspects such as security are left out by the facturers. Many of these devices work in portion of spectrum reserved internationally for industrial, scientific and medical band (ISM), where RF communications for short range could be vulnerables. In addition, the computational limitations of many of them make it difficult for them to have methods to strengthen their security level.

Wireless communications facilitate many activities in different environments, but it is also a gateway to malicious attacks through frequency spectrum analysis. The purpose of this work is to show the use of SDR and GNU Radio as potential tools to detect vulnerabilities in the communications of different IoT devices.

7.2. Objectives

The main objective of this work is to show the potential of SDR and GNU Radio to detect vulnerabilities in communications between IoT devices by analysing the electromagnetic waves of different protocols that use air as a transmission medium.

This proof of concept will be carried out through the following scenarios, which represent the specific objectives:

- Getting familiar with GNU Radio and SDR tools through the implementation of an FM receiver.
- Intercepting a 433 MHz signal for reverse engineering and a replay attack. This ISM band is widely used for short distance communications that require little energy consumption such as a home automation system, in telemetry systems for drones, among others.

7.3. Organization of work

The document is divided into six separate chapters, with the following in the subsequent chapters. The second chapter contains the state of the art. The third chapter summarize the theoretical information relevant to the application of the experimental environment. The fourth chapter describes the development of the project. The fifth chapter presents the analysis of the results. And finally, the sixth chapter mentions the conclusions and future work that could be done.

In accordance with the regulations, chapters 7 and 8 correspond to the translation into English language of the introduction and conclusions chapters.

Capítulo 8

Conclusions and future work

8.1. Conclusions

After gathering information during the development of this work, it can be concluded that although the concept of Software Defined Radio is not new, it is a hardware that can adapt and work with different transmission formats. Additionally, the prominence that IoT has gained in everyday life and in different sectors has made this tool, together with GNU Radio, an excellent alternative for the analysis of different communication protocols such as RF, Bluetooth and WiFi, and others.

As the number of available IoT options increases, so does the possible gap security of them. IoT devices operating in the ISM bands for sub-gigahertz frequencies are more sensitive as they can not only intercept the transmission but also compromise the integrity of the communication by impersonating a device.

Performing the tests outlined in this work was a challenge since the expected result was not achieved on the first attempt but required multiple repetitions. But, the fact that SDR and GNU Radio are open source tools and have an excellent community that is ontinuously being updated was of great help.

While it is true that the SDR - GNU Radio can be used to breach RF connections, it can also provide the option of prototyping or ethical hacking that allows the development of protection mechanisms to strengthen security as a defence mechanism against the various

attacks posed.

8.2. Future Work

The work done can be scaled up to a real environment simply because we have this type of communication within reach. As future work it is proposed to take into account the following considerations:

- The decoding method used in the proof of concept is effective but not very efficient. In fact, if the transmission contains a considerable number of packets, it would be somewhat complicated to decode the signal. For this reason, it would be interesting to automate this process so that the same program identifies the level corresponding to a logical 1 and 0 and thus achieve a more expeditious decoding.

- Through various contributions made on the vulnerability of different IOT technologies, develope new methods of protection against new threats that may emerge.

Bibliografía

[1] Rtl-sdr blog v3 datasheet. URL: https://www.rtl-sdr.com/wp-content/uploads/2018/02/RTL-SDR-Blog-V3-Datasheet.pdf (Accedido en Julio, 2020).

[2] DEFCONConference (2015). DEF CON 23 - Samy Kamkar - Drive it like you Hacked it: New Attacks and Tools to Wireles. URL: https://www.youtube.com/watch?v=UNgvShN4USU&t=1464s (Accedido en Junio,2020).

[3] GNU Radio (2017). GRCon17 - Radio Exploitation 101 - Matt Knight, Marc Newlin. URL: https://www.youtube.com/watch?v=xtjY_i6k5XQ (Accedido en Junio,2020).

[4] RSA Conference (2019). RF Exploitation: IoT and OT Hacking with Software-Defined Radio. URL: https://www.youtube.com/watch?v=88RfClJvPRQ (Accedido en Junio,2020).

[5] EMMANUEL MENDOZA ACEVEDO. *IMPLEMENTACIÓN DE UN SISTEMA DE CAPTURA DE PAQUETES EN REDES INALÁMBRICAS 802.11 Y BLUETOOTH.* PhD thesis, Universidad Tecnológica de la Mixteca, 2005.

[6] Víctor Aguado de Astorza. *Análisis de la seguridad y proceso de la auditoría de señales.* PhD thesis, 2017.

[7] Fahmida Ahmed, Shakh Md Alimuzjaman Alim, Md Shafiqul Islam, Kanti Bhusan Roy Kawshik, and Shafiul Islam. 433 mhz (wireless rf) communication between two arduino uno. *American Journal of Engineering Research (AJER)*, 5(10):358–362, 2016.

[8] Iván Barneda Faudot et al. Zigbee aplicado a la transmision de datos de sensores biomedicos. 2008.

[9] K. Chang. Bluetooth: a viable solution for iot? [industry perspectives]. *IEEE Wireless Communications*, 21(6):6–7, 2014.

[10] Trabajo Fin and D E Grado. Análisis de la Seguridad y Proceso de la Auditoría de Señales. Technical report, 2016.

[11] Elvis Eduardo Gaona García, Juanita Rodríguez Garay, and Camilo Humberto Florez Contreras. Modelamiento y simulación de las etapas de modulación digital y acceso al medio para un satélite de órbita baja. *Ingeniería*, 13(1):52–57, 2008.

[12] Antonio Jesús González García. Iot: Dispositivos, tecnologías de transporte y aplicaciones. 2017.

[13] José Javier Anguís Horno. Redes de área local inalámbricas: Diseño de la wlan de wheelers lane technology college. *Universidad de Sevilla*, 2008.

[14] ING SANTIAGO MANZANO and VALENCIA LLERENA CARLOS ANDRES. Hacking ético al iot mediante sdr. 2018.

[15] Juan Pablo Montero Hidalgo. Implementación de un sistema de comunicaciones basado en software radio. B.S. thesis, 2014.

[16] Arduino Nano. Arduino nano. URL: https://store.arduino.cc/usa/arduino-nano (Accedido en Agosto, 2020).

[17] Kiara Navarro, Francisco Canto, and PhD Héctor Poveda. Software defined radio as an educational learning tool in wireless communications.

[18] Carlos Lorenzo Nina Choque. *Sistema de telecontrol de una residencia por medio de un módulo inalambrico RF*. PhD thesis, 2011.

[19] Lucas F Payes, Oscar G Lombardero, and Víctor J Toranzos. Diseño de un transmisor de datos con modulación fsk en banda hf para un sistema de telemetría. *Extensionismo, Innovación y Transferencia Tecnológica*, 6:240–249, 2020.

[20] Randy Verdecia Peña. Análisis del desempeño de los esquemas de modulación bpsk y qpsk para diferentes condiciones de canales en sistema gfdm. *Maskay*, 8(1):7–12, 2018.

[21] J-M Picod, Arnaud Lebrun, and J-C Demay. Bringing software defined radio to the penetration testing community. In *Black Hat USA Conference*, 2014.

[22] Martin Pozniak, Debanjan Sadhukhan, and Prakash Ranganathan. Rf exploitation and detection techniques using software defined radio: A survey. In *2019 IEEE International Conference on Electro Information Technology (EIT)*, pages 345–350. IEEE, 2019.

[23] Joaquín Luque Rodríguez and Sebastián Clavijo Suero. Modulación de señales digitales. *Universidad Politécnica de Sevilla, Departamento de Tecnología Electrónica*, 1995.

[24] P Rodriguez, Juan P Villar, C Tarin, and S Blásquez. *Sociedad Digital en España 2019*. 2020.

[25] Štefunko Simon. Honeypot pro bezdrátové iot sítě. 2019.

[26] BANDAS ISM SUB-GIGAHERCIO. Grado en ingeniería de tecnologías y servicios de telecomunicación.

[27] UDC. Worldwide wearables market braces for short-term impact before recovery in 2020, according to idc. URL: https://www.idc.com/getdoc.jsp?containerId=prUS46138520 (Accedido Agosto,2020).

[28] Qing Yang and Lin Huang. *Inside Radio: An Attack and Defense Guide*. Springer, 2018.

Apéndice A

Códigos Utilizados

El siguiente enlace https://github.com/elizabethrivera/TFM_GNRadio_SDR.git/ contiene los ficheros de código generados por GNU Radio para los diferentes flujogramas empleados y el código de los micontroladores. El mismo tiene la siguiente estructura:

Receptor FM

- FM_Receiver.py
- FM_receiver2.grc

Captura Señal 433 MHz

- FM_Reversing_radio_signals.py
- receptor_433_hackrf.grc: almacena la señal capturada en formato IQ.
- reversing1.grc: almacena la señal capturada en formato wav.

Transmisor Señal 433 MHz

- transmisor_hrf.py
- transmisorHRF.grc:

Arduino

- Tx.ino: código para transmitir.
- Rx.ino: código para recibir.